National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

University of Alberta

The Application of Bond Graph Techniques to Robotic Manipulators

by

Dan Bright

A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree
of Master of Science

Electrical Engineering
Edmonton, Alberta
Fall 1994

National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-95009-9

Canada

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Dan Bright
TITLE OF THESIS: The Application of Bond Graph Techniques to Robotic Manipulators

DEGREE: Master of Science
YEAR THIS DEGREE GRANTED: 1994

Permission is hereby granted to the UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication rights and other rights in association with the copyright of the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

(Signed) Dan Bright

Permanent Address:
16B Meadowlark Village,
Edmonton, Alberta,
Canada

Date: July 7, 1994

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **The Application of Bond Graph Techniques to Robotic Manipulators** submitted by **Dan Bright** in partial fulfillment of the requirements for the degree of Master of Science.

V. G. Gourishankar

Dr. V. G. Gourishankar (Supervisor)

Dr. Q. H .Meng

W. S. Young

Dr. W. S. Young

Date: June 30 1994

To Natalia

Abstract

In this thesis, the relationship between bond graph techniques, and those conventionally used to study manipulator dynamics, is explained. Utilizing the same kinematic structure used to obtain the Newton-Euler formulation of the link dynamics, a multibond graph link model, suitable for use as a component in an object-based modeling environment, is created.

A SIMULINK block-diagram model of a revolute manipulator link, based on the multibond graph link model, is constructed. The SIMULINK model is then used as a component to simulate a planar two-link manipulator.

Acknowledgements

I would like to express my thanks and appreciation to my supervisor, Dr. V. G. Gourishankar for his support and guidance throughout this project. In addition, I would like to thank Dr. Q. H. Meng and Dr. W. S. Young for agreeing to sit on my defence committee.

For the classes. coffee, and doughnuts we shared together, I would like to thank my fellow graduate student, Hassan Farzadeh. Also, I would like to thank my office mate Sherman Chen, and my undergraduate lab partner Alex Nip for their support and friendship.

The financial support for this work should not go unmentioned, as the project would not have been possible without it. The Department of Electrical Engineering, University of Alberta, provided financial support through teaching assistantships, and the Natural Science and Engineering Research Council of Canada (NSERC) provided support through operating and equipment grants to my supervisor.

Contents

1 Introduction **1**

1.1 Introduction 1

1.2 Bond Graph Modeling of Mechanisms 3

1.3 Thesis Organization 4

2 Bond Graph Fundamentals **6**

2.1 Introduction 6

2.2 The Bond Graph Language 7

2.3 Port Variables 9

2.4 Multiport Characteristics 10

2.4.1 One-Ports 10

2.4.2 Two-Ports 12

2.4.3 Three-Ports 13

2.5 Application of Bond Graphs 13

2.5.1 Electrical Systems 14

2.5.2 Mechanical Systems 16

2.6 Obtaining Equations From Bond Graphs 22

2.6.1 Causality 22
2.6.2 Augmented Bond Graphs 25
2.6.3 Obtaining the State Equations 26
2.7 Summary 28

3 Applying Bond Graphs To Rigid Manipulators **29**
3.1 Introduction 29
3.2 Bond Graphs and Redundant Coordinates 31
3.3 Accelerated Reference Frames 32
3.4 Multibond Graphs 34
3.5 Link Numbering Scheme and Notation 37
3.6 Link Kinematics 37
3.6.1 Angular Velocity 38
3.6.2 Linear Velocity 38
3.7 Multibond Graph Link Model 41
3.8 Example: Single Link Manipulator 43
3.9 Obtaining the N-E Equations From the Multibond Graph Link Model 48
3.10 Summary 54

4 Simulating Multibond Graphs with SIMULINK **55**
4.1 Introduction 55
4.2 Simulation and Derivative Causality 58
4.3 A SIMULINK MTF Block 59
4.4 A SIMULINK Link Model 61

4.5 Example: Two-link Planar Manipulator 66
4.5.1 SIMULINK Model Construction 67
4.5.2 Simulation Results 71
4.6 Summary 74

5 Summary and Conclusions **75**
5.1 Summary 75
5.2 Conclusions 76
5.3 Suggestions For Future Research 77

A S-Functions for MTF Block **80**
A.1 Connections, Appearance, and Masking for MTF Block 80
A.2 Internal Function Called by MTF Block 89

B Equations From Augmented Multibond Graph Link Model **92**

C S-Function for Two-link Manipulator **94**

Bibliography **95**

List of Figures

1 An R-L-C Circuit and its Corresponding Bond Graph 7
2 Simple Bond Graphs 7
3 Typical Multiport Elements 8
4 Oriented Bond Graphs 8
5 Tetrahedron of States 9
6 Bond Graph of R-L-C Circuit 15
7 Bond Graph of Resistor Network 17
8 Bond Graph of Mass–Spring–Damper System 19
9 Bond Graph of Rack and Pinion System 21
10 Interpretation of Causal Stroke 23
11 Causality for Bond Graph Elements 26

12 Bond Graph Model of a Physical Pendulum 32
13 Bond Graph Model of a Euler's Equations 34
14 Multibond Graph Model of a Euler's Equations 36
15 Multibond Graph Junction Structure for Link Angular Velocity 38
16 Three Coordinate Frames 40
17 Multibond Graph Link Junction Structure 41

18 Multibond Graph Link Model 42
19 Single-Link Manipulator 43
20 Single-Link Manipulator Multibond Graph 44
21 Augmented Single-Link Manipulator Multibond Graph 44
22 Newton-Euler Multibond Graph Link Model 49

23 First part of MTF S-function 61
24 Dialog for entering MTF parameters 62
25 Augmented Multibond Graph Model 63
26 A Part of the Link Block Diagram 63
27 Single Link Block Diagram 64
28 SIMULINK Link Block Diagram 65
29 SIMULINK Link Component Block 66
30 Two-link Planar Manipulator 67
31 First Stage of Model Construction 68
32 SIMULINK Model of Two-link Planar Manipulator 71
33 Graph of Link Angular Velocity 73

List of Tables

2.1 Summary of One-Port Elements 10
2.2 Summary of Two-Port Elements 13
2.3 Summary of Three-Port Elements 14
2.4 Causality for Bond Graph Elements 24

4.5 Results of Varying Maximum and Minimum Step Sizes 72

Chapter 1

Introduction

1.1 Introduction

A robotic manipulator is a complex mechanical assemblage of joints, links, and actuators, coordinated by an active or passive control system. The inspiration for creating robotic manipulators comes from living organisms. That is, the goal of robotics design is to create mechanical systems that perform as capably as biological systems.

Robots are able to operate in environments that are hostile to living organisms; this appears to be their main advantage. In terms of adaptability, robustness, and even dynamic performance, the current generation of robotic manipulators compares poorly with corresponding biological systems. For example, a typical industrial manipulator has a maximum payload capacity that is twenty times smaller than the mass of the manipulator itself. In contrast, the human arm is known to lift payloads that are twice its own

mass. A consequence of the industrial manipulator's much smaller payload–to–mass ratio is a degradation in speed; manipulators that have a payload capacity comparable to that of the human arm operate more slowly[20, p. xxii].

In order to improve the capabilities of robotic manipulators, the design problem has been decomposed into a number of simpler problems and associated research areas. For a classical robot architecture, areas of research include: control; vision; sensor fusion; path planning; as well as mechanical design[17].

While decomposition of the robotics design problem is necessary, it also introduces a difficulty of its own: research carried out in one of the specialized areas may be so specific that it cannot be integrated into a working robot design[4]. One reason this problem arises is that the design and construction of robotics manipulators is a time-consuming and expensive process. Thus, the researcher is faced with simulating the robotic manipulator. If the simulation is constructed *from scratch*, it is quite possible that invalid assumptions may have to be made about those aspects of the robotic manipulator with which the researcher is less familiar.

It has been proposed that the verification of research results, through simulation, can be improved significantly if an object-based approach to model construction is adopted. Using this approach, a model of the robotic manipulator is built by connecting a number of component models together. If the component models are closely related to the physical components of the system, the construction of the robot model is essentially a schematic for constructing the physical system[19]. Further, the robot modeler is able to

treat components as *black boxes*, needing only to know how the components are connected, not the details of how they work.

Presently, there is no common modeling language for robotic manipulators, apart from mathematical equations. However, advocates of bond graph modeling techniques have suggested that bond graphs constitute an ideal graphical language for the object-based modeling of complicated dynamics systems, such as robotics.

1.2 Bond Graph Modeling of Mechanisms

Bond graph techniques provide a graphical language for modeling dynamic systems[21]. These techniques model system dynamics at a higher level than mathematical equations. The advantage of this is that the components that comprise a physical system can be readily identified in a bond graph model.

Mechanical engineers were among the first users of bond graph techniques [12]. For this reason, a large body of literature exists on the application of bond graph techniques to multi-body dynamic systems, such as robotics.

Shahinpoor [23] presented bond graph modeling as an alternative to Newton-Euler and Lagrange approaches. However, the large number of coordinates used to describe multi-body systems makes the resulting bond graph unwieldy. For this reason, most researchers have adopted a vector-based extension to the bond graph, called the multibond graph, described in [6, 7, 8].

Allen presents a multi-bond graph approach to modeling links in accelerated reference frames [1]. This paper uses center-of-mass coordinates, and Euler angles to describe the geometric configuration of the links. A survey of

additional papers published on the application of multibond graphs to multibody systems is presented in [12]. Although a number of these papers use robotic systems as examples, they do not adopt the coordinate assignment conventions used in robotics. Thus, it remains hard to relate the multibond graph techniques to those conventionally used to obtain the dynamic model of a robotic manipulator.

1.3 Thesis Organization

In this thesis, bond graph techniques are applied to the dynamics of robotic manipulators. Through the development of a bond graph link model, suitable for use with object-based modeling techniques, the relationship between bond graph modeling techniques and those conventionally used for robot dynamics is explained. An example of using the component link model, with the computer simulation program SIMULINK, is provided.

- Chapter 2 provides a tutorial on the use of bond graphs. Bond graph techniques are applied to simple mechanical and electrical systems; a systematic method for obtaining equations from a bond graph model is reviewed.

- In chapter 3 the bond graph concept is extended to include multibonds. Multibond graph techniques are then applied to develop the dynamic model for a robotic link. To verify the correctness of this model, the Newton-Euler equations for robot dynamics are obtained from the multibond graph link model.

- The simulation of a robotic manipulator using the simulation program SIMULINK is presented in chapter 4. A component block diagram model, based on the multibond graph link model, is developed. The use of the component block diagram to construct the manipulator model is presented.

- Chapter 5 presents a brief summary and conclusion, and suggests topics for future research.

- Appendix A contains the two SIMULINK S-files used to implement the SIMULINK modulated transformer block, described in chapter 4.

- In appendix B, the equations used to implement the SIMULINK link block diagram (see chapter 4) are presented.

- A listing of the SIMULINK S-file, that implements the two-link manipulator example presented in chapter 4, is given in appendix C. Due to the large size of this file, appendix C is contained in a supplemental volume.

Chapter 2

Bond Graph Fundamentals

2.1 Introduction

Bond graphs, created by Henry Paynter in 1959, constitute a graphical language for describing lumped–parameter dynamic systems in terms of energy and information flow [21]. The bond graph language was developed to model engineering systems that contain more than one type of energy; thus systems containing hydraulic, mechanical, electrical, or magnetic energies, among others, are all modeled using a uniform notation and a small set of ideal elements.

Bond graph techniques were developed as a generalization, in terms of notation, of the electrical network concept [12]. In fact, any advantage that bond graphs have over equivalent electrical networks is qualitative; for every ideal bond graph element there is a corresponding ideal circuit element. This leads to a difficulty in justifying a preference for the use of bond graphs.

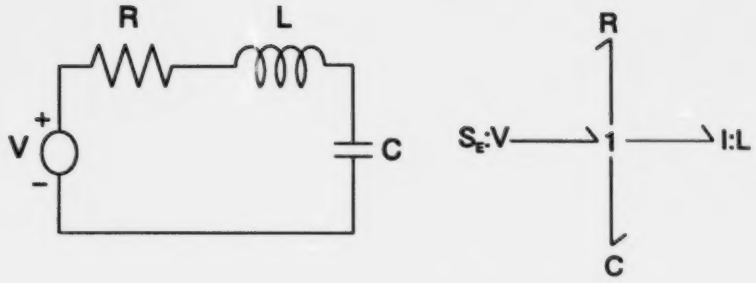


Figure 1: An R-L-C Circuit and its Corresponding Bond Graph

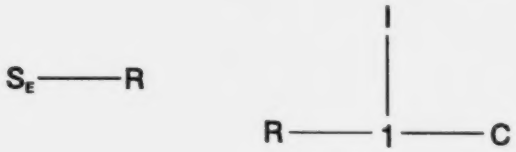


Figure 2: Simple Bond Graphs

Visually, however, bond graphs and electrical networks appear significantly different. This difference is illustrated in figure 1 which shows a simple R–L–C circuit and its corresponding bond graph. It is now recognized that bond graphs present dynamic systems in a new light, leading to new insights into their behaviour. As can be seen from figure 2, which shows some simple bond graphs, the name *bond graph* was arrived at by observing the similarity between these graphs and chemical bond diagrams.

2.2 The Bond Graph Language

Bond graphs model physical systems as a collection of interacting elements. Interaction occurs when power is transmitted from one element to another

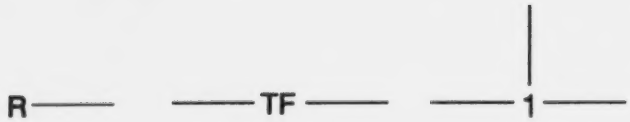


Figure 3: Typical Multiport Elements

through ports identified with the elements. Since each element has one or more ports associated with it, the term *multiport* is used to describe the class of all elements that comprise a bond graph [21]. Examples of typical multiports are shown in figure 3. As shown in the figure, multiports are designated with alphanumeric characters, and the connecting ports are represented by short line segments.

The connection of ports between pairs of multiports, to facilitate interaction, is called *bonding*. A bond graph is defined as a collection of multiports, bonded together [21].

To model physical systems with bond graphs, it is necessary to indicate the direction of power flow within the graph. This is accomplished by attaching a half arrow to each bond in the graph, indicating the direction of positive power. Such a bond graph is said to be *oriented* [21]. Figure 4 shows examples of oriented bond graphs.

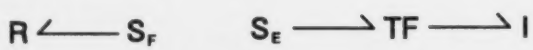


Figure 4: Oriented Bond Graphs

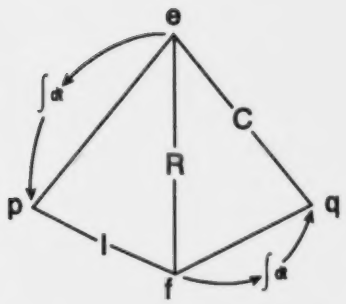


Figure 5: Tetrahedron of States

2.3 Port Variables

Each connecting port has, associated with it, two power variables: effort and flow. Both of these variables are assumed to be scalar functions of an independent time variable, t. Power is found directly from the product of the effort and flow:

$$P(t) = e(t)f(t)$$

Momentum $p(t)$ and displacement $q(t)$ can be found as the time integrals of the effort and flow variables respectively. The relationship between port variables is often illustrated using Paynter's *tetrahedron of states*, shown in figure 5 [22, p 28].

The vertices of the tetrahedron are labeled with the four port variables: e, p, f,and q. Each edge of the tetrahedron represents a possible relationship between the port variables. The edges labeled I, R, and C represent passive one-port relationships, explained in section 2.4.1.

Name	Defining Relationship		Symbol
	General	Linear	
Effort Source	e=e(t)	e=e(t)	S_E ⟶
Flow Source	f=f(t)	f=f(t)	S_F ⟶
Resistance	e=ϕ(f)	e=Rf	R ⟵
Capacitance	q=ϕ(e)	q=Ce	C ⟵
Inertance	p=ϕ(f)	p=If	I ⟵

Table 2.1: Summary of One-Port Elements

2.4 Multiport Characteristics

Bond graphs are composed using only a few basic types of multiports. Each multiport is classified by the number of ports it possesses, as well as its energy characteristics. A multiport that has n power ports is referred to as an *n-port*. In terms of energy characteristics, multiports can be active or passive, depending on whether or not they are a source of energy. Only nine types of multiports are needed to model lumped-parameter dynamic systems: five one-ports, two two-ports, and two three-ports.

2.4.1 One-Ports

The simplest type of multiport is the one-port. Of the five basic one-ports, two are active and three are passive. The active one-ports serve as sources of effort and flow, while the passive elements are used to store or dissipate energy. A summary of the basic one-ports showing name, defining relationship, and symbol is shown in table 2.1.

Active One-Ports

The *effort source*, denoted S_E, is defined by the relationship $e = e(t)$, and the *flow source*, S_F, is defined by the relationship $f = f(t)$. Both of these sources are ideal, and equivalent to the ideal voltage and current sources used in electrical circuits. Because these elements are sources of power, the half-arrows that indicate the direction of power flow, always point away from the source.

Passive One-Ports

The passive one-ports, resistance (R), capacitance (C), and inertance (I), define mathematical relationships between the port variables. As the name *resistance* implies, the resistance one-port dissipates energy. In contrast, the inertance and capacitance one-ports store energy.

The resistance one-port (R) relates an effort and flow variable. This can be written as $e = \Phi(f)$, generally, or as $e = Rf$ if the relationship is linear.

The capacitance one-port defines a relationship between a displacement variable (the time integral of the flow variable), and an effort variable. This can be expressed as $q = \Phi(e)$. If the relationship between q and e is linear, the expression may be rewritten as $q = Ce$.

The last passive one-port, inertance, is defined by the relationship $p = \Phi(f)$, which relates a momentum variable (the time integral of the effort) to a flow variable. For linear relationships, the expression may be rewritten as $p = If$.

The relationships defined by the passive one-ports are shown graphically

in the tetrahedron of states (figure 5). To indicate that they dissipate or store energy, the half-arrows associated with the passive one-ports always point towards the element.

2.4.2 Two-Ports

For two-ports, only those elements that conserve power are considered. As mentioned earlier, only two such elements are required.

The first two-port, called the *transformer* (TF), is defined by the relationship $e_1 = me_2$ and $mf_1 = f_2$. In this relationship, the variable m may either be constant, or an arbitrary function, without affecting the power-conserving properties of the the two-port. In those cases where m is not constant, the element is referred to as a *modulated transformer* (MTF).

The second two-port is the *gyrator* (GY). This element is defined by the relationships $e_1 = mf_2$ and $mf_1 = e_2$. Once again, m may be constant or an arbitrary function. In the latter case, the element is called a *modulated gyrator* (MGY).

Sign conventions for transformers and gyrators have one half-arrow pointing towards the element, and one half-arrow pointing away from the element. This marking indicates that power flows through the two-port.

A summary of the two-port elements, showing name, symbol, and defining relationship, is shown in figure 2.2.

Name	Defining Relationships		Symbol
Transformer	$e_1 = me_2$	$mf_1 = f_2$	⟶TF⟶
Gyrator	$e_1 = mf_2$	$mf_1 = e_2$	⟶GY⟶

Table 2.2: Summary of Two-Port Elements

2.4.3 Three-Ports

As with two-ports, only elements that conserve power are used. The two elements used are the *1-junction* and the *0-junction.*

The 1-junction is defined by the pair of relationships: $e_1 + e_2 + e_3 = 0$, and $f_1 = f_2 = f_3$. In electrical systems, where effort is identified as voltage and flow is identified as current, it can be seen that the definition of the 1-junction simply restates Kirchoff's voltage law [16, pp. 34–37]. In the same vein, the 0-junction represents Kirchoff's current law and is defined by the relationships: $e_1 = e_2 = e_3$, and $f_1 + f_2 + f_3 = 0$.

Unlike one-ports and two-ports, signs for three-ports may be difficult to assign. The easiest approach is to determine the signs from the actual system under study, so that consistency with the signs of attached one-ports and two-ports can be maintained. The properties of the three-ports are summarized in table 2.3.

2.5 Application of Bond Graphs

The application of bond graphs to engineering systems is best presented by way of example. For simplicity, attention is restricted to the development and

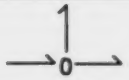
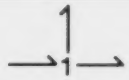
Name	Defining Relationships		Symbol
0–Junction	$e_1=e_2=e_3$	$f_1+f_2+f_3=0$	
1–Junction	$f_1=f_2=f_3$	$e_1+e_2+e_3=0$	

Table 2.3: Summary of Three-Port Elements

solution of bond graphs for mechanical and electrical systems. This does not mean that bond graphs are more difficult to apply to other energy domains. Electrical and mechanical systems have been chosen because they are most relevant to robotics.

2.5.1 Electrical Systems

In electrical circuits, effort is interpreted as voltage, while flow is interpreted as current. Bond graph elements – resistance, capacitance, and inertance – are used to represent resistance, capacitance, and inductance (respectively) in the electrical circuit. Since 0-junctions have a constant effort, they represent parallel connections within the circuit. 1-junctions, having a constant flow, represent series connections.

R-L-C Circuit

To develop the bond graph for the simple series R-L-C circuit, shown in figure 6a, the first step is to identify the junction structures. In this case there is only a single 1-junction that represents the series connection of all the circuit

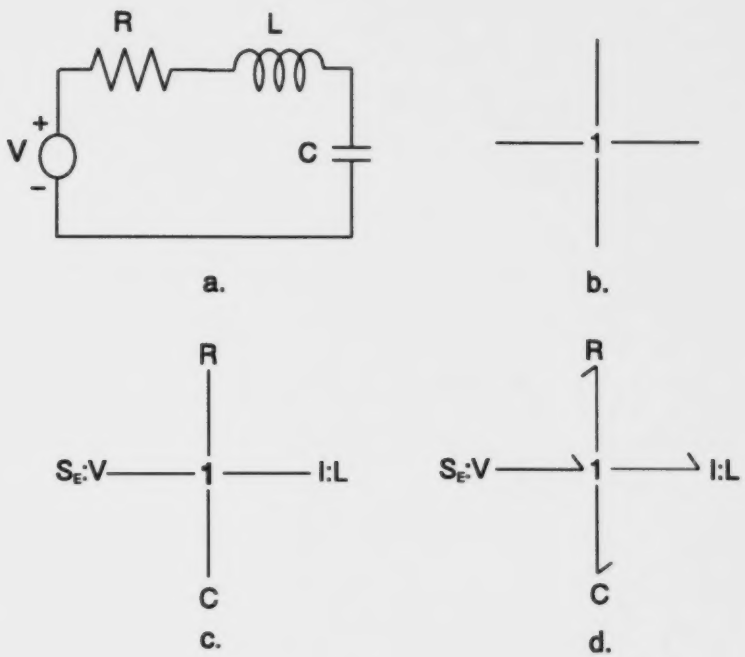


Figure 6: Bond Graph of R-L-C Circuit

elements.

The next step is to attach all elements that share a common flow with the 1-junction. Here there are four circuit elements, but only three ports available on the 1-junction. This is rectified by bonding an additional 1-junction onto the existing one. Generally, the bond between the new junction pair is dissolved, and the new junction is drawn as shown in figure 6b. Manipulating junction structures in this manner is mathematically consistent, and shows why there is no need for basic multiports having more than three ports.

Having attached the one-ports to the junction (figure 6c), half-arrows are attached to the bonds, consistent with the sign conventions of the one-ports.

The resulting oriented bond graph is shown in figure 6d.

Resistor Network

As a second example of the application of bond graphs to electrical circuits, the resistor network shown in figure 7a is considered. Once again, the first step is to identify the junction structures. The two nodes, identified in the figure, become 0-junctions, while the paths between the nodes become 1-junctions. The junction structure is shown in figure 7b.

The next step is to attach the resistive elements to the 1-junctions, as shown in figure 7c. If it is assumed that node 2 is grounded, the corresponding 0-junction may be eliminated. This leads to the bond graph shown in figure 7d. A further simplification can be made by eliminating the 1-junction to which R_3 is attached.

To orient the bond graph, half-arrows are placed on the bond connected to one-ports. As shown in figure 7e, this still leaves two bonds that are not oriented. Summing the current at node 1 of the circuit gives: $I_3 = I_1 + I_2$. Applying this information, the completed bond graph is shown in figure 7f.

2.5.2 Mechanical Systems

Unlike low frequency electrical circuits, which can be described by their topology alone, mechanical systems have geometrical constraints. Thus, in addition to the dynamics of a mechanical system, the kinematics must also be considered. The kinematics of a mechanical system can be identified as the linear graph defined by the two-ports and three-ports that comprise the sys-

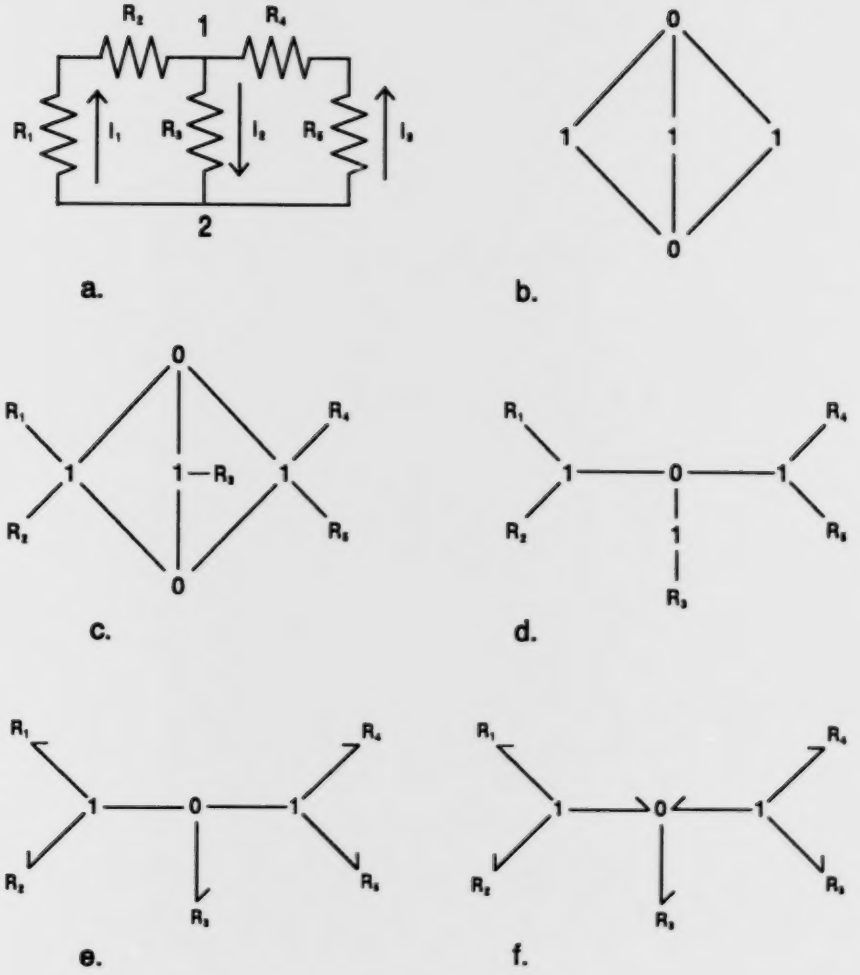


Figure 7: Bond Graph of Resistor Network

tem bond graph.

For mechanical systems, two cases are considered:

- For systems undergoing pure translation, the effort variable is force, and the flow variable is velocity. The resistance one-port is interpreted as friction; the capacitance one-port becomes a spring; and the inertance one-port is interpreted as mass.

- For systems undergoing pure rotation, the effort variable is torque, while the flow variable is angular velocity. In this case, the resistance one-port is interpreted as rotating friction; the capacitance one-port becomes a rotational spring, and the inertance element represents the moment of inertia.

Mechanical Mass–Spring–Damper System

As a first example in using bond graphs for mechanical systems, consider the mechanical system shown in figure 8a. This system is the mechanical analog to the electrical R-L-C circuit of figure 6. It is interesting to see that a series connection in an electrical circuit corresponds to a parallel connection in a mechanical system.

Once again, the first step towards developing the system bond graph involves identifying the junctions. In this case there is a single 1-junction that corresponds to the velocity $\dot{x}$. Next the one-ports are attached to the junction, as shown in figure 8c. Note that the one-ports have additional information associated with them, i.e., each of the one-ports is labeled with

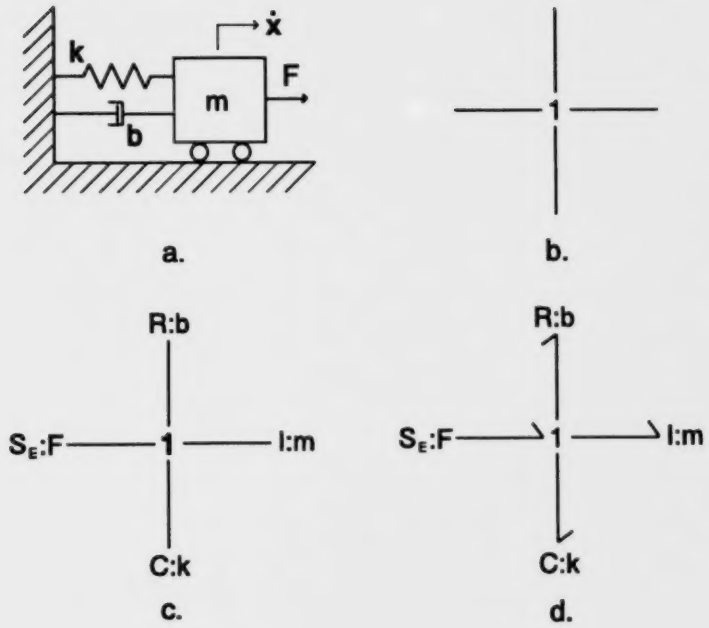


Figure 8: Bond Graph of Mass–Spring–Damper System

the corresponding parameters of the mechanical system. Finally, half-arrows are attached to the bonds, resulting in the oriented bond graph of figure 8d.

To explore this example further, consider what happens if the direction of F is reversed in figure 8a. The bond graph is no longer oriented properly. The direction of positive power is determined by the product of F and $\dot{x}$, and the half-arrows on the source one-ports indicate the direction of positive power. The change in direction of F can be indicated by labeling the effort one-port with $S_E : -F$.

Rack and Pinion

As a second application of bond graphs to mechanical systems, consider the rack and pinion assembly shown in figure 9a. In this case it is useful to consider three velocities: ω_1, ω_2, and $\dot{x}$. The rotational spring exerts an equal (but opposite) torque on both rotating masses proportional to the difference between ω_1 and ω_2. Thus, a common effort junction, to which the spring is attached, relates, ω_1 and ω_2. The relationship between $\dot{x}$ and ω_2 is obtained by taking the time derivative of the expression: $x = r\theta_2$. This relationship has the same form as the transformer two-port. Using these pieces of information, the junction structure shown in figure 9b is arrived at.

Figure 9c shows the partially oriented bond graph that results from attaching the one-ports, with their requisite signs, to the junction structure. This leaves only two bonds, connecting the 1-junctions to the 0-junction, unoriented. Orientation of these bonds is determined through the introduction of a relative velocity, ω_3, defined by the relationship: $\omega_3 = \omega_2 - \omega_1$. Alterna-

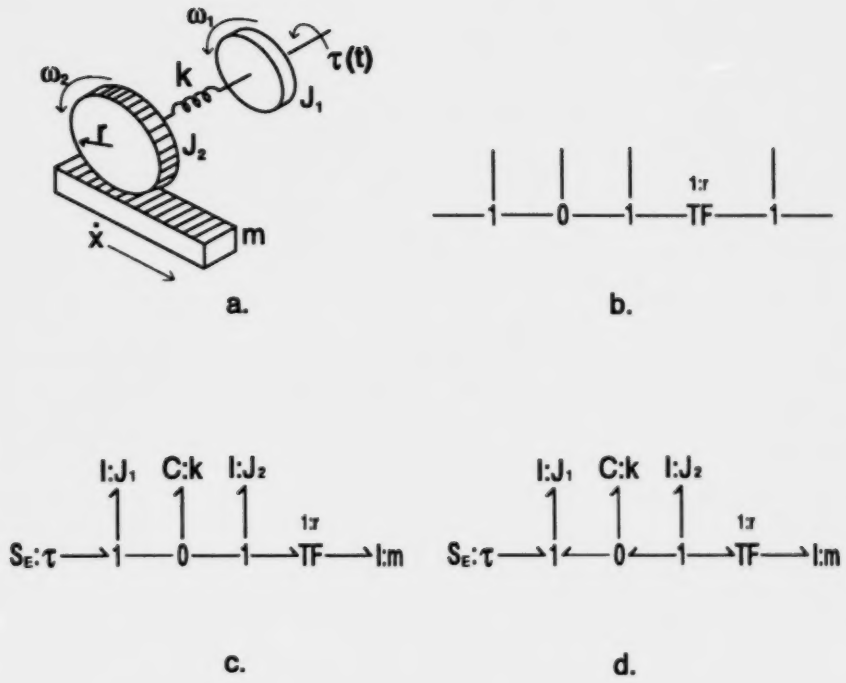


Figure 9: Bond Graph of Rack and Pinion System

tively, the relationship could be defined by: $\omega_3 = \omega_1 - \omega_2$. The importance of the relationship, between ω_1 and ω_2, lies in the magnitude of the difference; the sign of the difference is unimportant. The completed oriented bond graph is shown in figure 9d.

2.6 Obtaining Equations From Bond Graphs

A bond graph presents a mathematical model of a dynamic system in graphical terms. A set of differential equations, describing the system dynamics, can be obtained by writing down and combining the mathematical relationships found in a bond graph.

The process of generating equations from a bond graph can be greatly simplified if additional information is placed on the oriented bond graph. This information uniquely identifies the effort and flow variables associated with each bond in the graph. In addition, *independent* effort and flow variables are distinguished from those that are *dependent.* Bond graphs, to which this additional information has been added, are referred to as being *augmented.* Differential equations generated from an augmented bond graph are naturally in a state-space form.

2.6.1 Causality

Distinguishing between dependent and independent variables is accomplished by applying the concept of *causality* to the bond graph. Causality is a mathematical tool used as an aid in the generation of equations from the bond

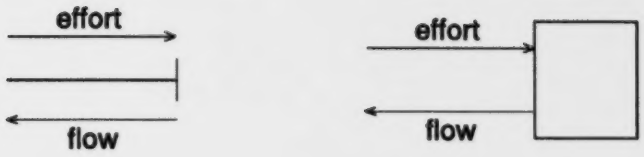


Figure 10: Interpretation of Causal Stroke

graph; it does not necessarily have physical significance. Each one-port in a bond graph defines a relationship between an effort variable and a flow variable. Causality, indicated by a vertical stroke on the bond, indicates which of the bond variables is independent and which is dependent. The convention used for causal strokes is shown in figure 10.

Depending on how the causal stroke is applied, effort and flow variables associated with an energy-storing one-port have either an integral or a derivative relationship. For example, if the flow variable associated with a capacitance one-port is independent, the relationship between the port variables is given by: $e = \Phi^{-1}(q)$. Since $f = \dot{q}$, the effort variable is a function of the integral of the flow variable. Note also that, with causality assigned in this manner, the variables e and f in the bond graph can be replaced with the variables q and $\dot{q}$.

When a bond variable is expressed in terms of the integral of the other bond variable, the bond is said to have integral causality. If all the bonds in a graph, connected to energy storing one-ports, are in integral causality then the equations describing the system dynamics can be written in terms of momentum $p(t)$ and displacement $q(t)$ variables, and their first derivatives, resulting in equations in state-space form.

Name	Possible Causality	
Effort Source	S_E ⊣	
Flow Source	S_F ⊢	
Resistance	R ⊢	R ⊣
Capacitance	C ⊢	* C ⊣
Inertance	* I ⊢	I ⊣
Trasnformer	⊢ TF ⊢	⊣ TF ⊣
Gyrator	⊢ GY ⊣	⊣ GY ⊢
1–Junction	⊢ 1	
0–Junction	⊣ 0	

*** Denotes Integral Causality**

Table 2.4: Causality for Bond Graph Elements

Application of causality to a bond graph may not always result in all bonds connected to energy storing one-ports having integral causality. Causality is propagated by the bond graph junction structure (two-ports and three-ports). When mixed integral and derivative causality occurs in a bond graph, this means that one or more of the system state variables is not independent. Possible causal assignments for the basic bond graph elements is shown in table 2.4.

To assign causality to a bond graph in a systematic fashion, the *sequential causality assignment procedure* (SCAP) from [22, p. 93] is presented:

1. Assign the requisite causality to all bonds connected to source elements

in the bond graph. Propagate the causal stroke as far as possible through the bond graph structure.

2. Choose any C or I element and assign integral causality. Again, propagate this causal stroke as far as possible through the bond graph structure. Repeat this procedure until all the C and I elements have been causally assigned.

3. Choose any R element that has not been assigned, and give it an arbitrary causality. After propagating the stroke through the bond graph structure, repeat this process until all bonds have been assigned.

2.6.2 Augmented Bond Graphs

In addition to placing causal strokes on the bond graph, it is convenient to number each bond in the graph and label it with an effort and flow variable. Bonds that terminate on energy storing one-ports, and are in integral causality, are labeled first using the derivative of the appropriate state variable.

For instance, the flow variable for bond n, terminating on a capacitance element and in integral causality, would be labeled with an effort variable, e_n, and a flow variable, $\dot{q}_n$. Similarly, the bond terminating on an inertance element in integral causality, would be labeled with the effort variable, $\dot{p}_n$, and the flow variable, f_n. Having done this, the system state variables can be identified from the bond graph as the momentum and displacement variables, p_n and q_n.

After identifying the system state variables, the remaining bonds are la-

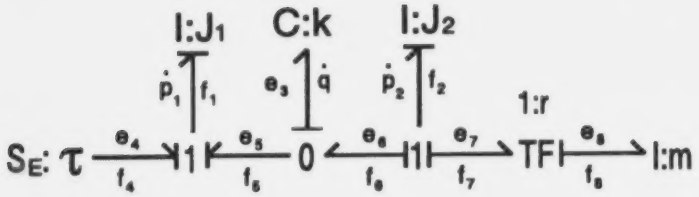


Figure 11: Causality for Bond Graph Elements

beled sequentially with effort and flow variables, e_i and f_i. An example of this procedure is shown in figure 11, which shows an augmented version of the bond graph for the rack and pinion system (see figure 9). In this example, both integral and differential causality are found. This *causal conflict* arises because the velocities ω_2 and $\dot{x}$ are not independent. In this case, the state variable could be associated with either of these two velocities. For convenience, the state variable associated with the angular velocity is used.

2.6.3 Obtaining the State Equations

With the aid of the augmented bond graph (11), the system state equations can be obtained by writing down and combining the mathematical relationships in the graph:

$$f_1 = \frac{p_1}{J_1} \tag{2.1}$$

$$e_3 = \frac{q}{K} \tag{2.2}$$

$$f_2 = \frac{P_2}{J_2} \tag{2.3}$$

$$\dot{p_1} = e_4 + e_5 \tag{2.4}$$

$$\dot{q} = f_6 - f_5 \tag{2.5}$$

$$\dot{p_2} = -e6 - e7 \tag{2.6}$$

$$e_3 = e_5 = e_6 \tag{2.7}$$

$$f_2 = f_6 = f_7 \tag{2.8}$$

$$f_1 = f_4 = f_5 \tag{2.9}$$

$$e_7 = re_8 \tag{2.10}$$

$$e_8 = m\dot{f}_8 \tag{2.11}$$

$$f_8 = rf_7 \tag{2.12}$$

Combining equations 2.3, 2.8, 2.11 and 2.12:

$$e_8 = mr\frac{\dot{p_2}}{J_2} \tag{2.13}$$

Combining equations 2.2, 2.6, 2.7 and 2.13:

$$\dot{p_2} = -\frac{q}{K} - mr^2\frac{\dot{p_2}}{J_2} = -\frac{J_2}{J_2 + mr^2}\frac{q}{K} \tag{2.14}$$

Combining 2.1, 2.3, 2.5, 2.8 and 2.9:

$$\dot{q} = \frac{p_2}{J_2} - \frac{p_1}{J_1} \tag{2.15}$$

Combining 2.4, 2.2, and 2.7:

$$\dot{p_1} = \tau + \frac{q}{K} \tag{2.16}$$

Finally, equations 2.14, 2.15, and 2.16 can be written in matrix form as:

$$\begin{bmatrix} \dot{p_1} \\ \dot{p_2} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{1}{K} \\ 0 & 0 & -\frac{J_2}{K(J_2+mr^2)} \\ -\frac{1}{J_1} & \frac{1}{J_2} & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ q \end{bmatrix} + \begin{bmatrix} \tau \\ 0 \\ 0 \end{bmatrix} \tag{2.17}$$

2.7 Summary

The bond graph modeling techniques, outlined in this chapter, present a highly organized approach to studying the dynamics of lumped-parameter engineering systems. The bond graph model of a dynamic system can be quickly constructed using a small set of ideal elements. The resulting model graphically represents the flow of power and energy within the system.

Once a bond graph model has been created, causality can be applied to it using SCAP, and a set of state-space equations can be obtained. The state variables correspond to the momentum and displacement variables of the system. The number, and type, of state variables for a particular dynamic system is readily visible in the system's augmented bond graph.

Bond graph techniques, as presented in this chapter, have been applied to robotic systems [23]. However, the large number of coordinates conventionally used to describe the position and orientation of a robotic system tends to make the corresponding bond graph too complicated to be practical, despite the simplicity of the notation. In the next chapter, extensions to the bond graph methodology, that overcome this problem, are introduced.

Chapter 3

Applying Bond Graphs To Rigid Manipulators

3.1 Introduction

For kinematic or dynamic analysis, a rigid manipulator can be viewed as a collection of rigid bodies, called links, connected serially by prismatic or revolute joints [9, p. 69]. Based on this view, a simple and general procedure for modeling manipulators can be formulated. First, a bond graph fragment[1] that represents a single link is created. The bond graph fragment can then be used as a component to construct the model for any serial manipulator.

The study of robot dynamics, as applied to rigid manipulators, is a well developed area described in a number of textbooks on robotics. The geometric configuration of the robot is typically described using the Denavit-

[1] A bond graph fragment is a graph containing free, or unconnected, ports

Hartenberg (D-H) notation [9, p. 74]. This provides an organized way to assign oriented coordinate systems to each joint of the manipulator.

Formulation of the system dynamics can be accomplished using either the Newton-Euler [9, p. 196] or Lagrangian dynamic approach [9, p.207]. Using either of these methods, the dynamic equations can be generated in a closed or iterative form.

The approach taken in this thesis – modeling manipulator dynamics from the point of view of the links – is essentially the same as that used by the Newton-Euler (N-E) method. In contrast, the Lagrange-Euler method develops the dynamics in terms of the kinetic and potential energies of the entire manipulator.

Before modeling a manipulator link, it is useful to extend the bond graph techniques presented in chapter 2 to include the concept of vectors. This produces an extremely compact notation that greatly simplifies the drawing of the link junction structure. An additional benefit of this extension is that it allows the similarity between the bond graph and N-E approaches to be maximally exploited; the bond graph junction structure can be drawn from the same kinematic (vector) equations that would be used for the N-E method.

3.2 Bond Graphs and Redundant Coordinates

The position of a robotic manipulator with n degrees of freedom (DOF) can be described completely using n independent coordinates. However, conventional approaches to robotic kinematics and dynamics attach an oriented coordinate system to each joint of the manipulator. Since each joint, considered independently, requires 6 variables to describe its position and orientation, $6n$ variables are used to describe the position of a manipulator. However, because each joint contributes only 1 DOF to the manipulator, $5n$ variables are redundant.

The use of redundant coordinates in a bond graph may lead to ambiguity in the interpretation of inertial parameters that are associated with angular velocities, i.e. J parameters. To overcome this problem, it is useful to restrict the use of J parameters to describe the moment of inertia about center of mass (CM) coordinates only. To illustrate this, consider the physical pendulum, shown in figure 12a. To analyze the dynamics of the physical pendulum, the distributed mass is replaced by a point mass located at the pendulum CM, a distance l from the pivot. Once this is done, it is clear that the angular velocity of the pendulum, about its CM, is the same as the angular velocity about its fixed point. The inertial parameter, J, can then be identified as being the moment of inertia about the pendulum CM, denoted J_{CM}. Using the parallel axis theorem, the moment of inertia about the pendulum pivot is: $J = J_{CM} + ml^2$. The corresponding bond graph is

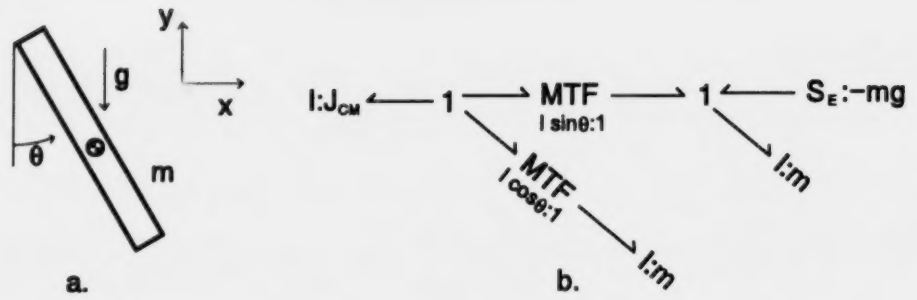


Figure 12: Bond Graph Model of a Physical Pendulum

shown in figure 12b.

Restricting the use of inertial parameters to CM coordinates implies that CM coordinates must be used, in addition to the joint coordinates, when developing the bondgraph junction structure for a robotic manipulator. Again, this approach is consistent with the N-E method.

3.3 Accelerated Reference Frames

Since the majority of industrial manipulators use rotational joints, coordinate frames[2] attached to the joints are accelerated with respect to an inertial base reference system. Under these conditions, the vector operator

$$\left(\frac{d}{dt}\right)_s = \left(\frac{d}{dt}\right)_r + \omega\times \tag{3.1}$$

can be used to relate the time derivative in the rotating reference frame with the time derivative in the inertial frame [15, p. 176]. Here $\left(\frac{d}{dt}\right)_s$ denotes the

[2]The term *frame* refers to a set of four vectors that describe position and orientation.

time derivative in the inertial frame; $\left(\frac{d}{dt}\right)_r$ indicates the time derivative with respect to the rotating frame; and $\boldsymbol{\omega}$ is the angular velocity of the rotating frame relative to the inertial frame.

As an example of incorporating accelerated reference frames into bond graphs, the bond graph for Euler's equations is considered. In an inertial reference frame attached to the CM of the rigid body, the angular momentum vector for the rigid body is given by:

$$\mathbf{H} = \boldsymbol{J\omega} \tag{3.2}$$

where $\boldsymbol{\omega}$ is the angular velocity vector of the rigid body, and $\mathbf{J}$ is a diagonal inertia matrix (tensor). The net torque acting on the body is then:

$$\boldsymbol{\tau} = \left(\frac{d\mathbf{H}}{dt}\right)_s$$

For a set of axes fixed in the rigid body this becomes:

$$\boldsymbol{\tau} = \left(\frac{d\mathbf{H}}{dt}\right)_r + \boldsymbol{\omega} \times \boldsymbol{H}$$

or

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} \dot{H}_1 \\ \dot{H}_2 \\ \dot{H}_3 \end{bmatrix} + \begin{bmatrix} \omega_2 H_3 - \omega_3 H_2 \\ \omega_3 H_1 - \omega_1 H_3 \\ \omega_1 H_2 - \omega_2 H_1 \end{bmatrix} \tag{3.3}$$

The bond graph corresponding to equation 3.3 is shown in figure 13. The junction structure of this bond graph, a modulated gyrator ring, is referred to as an Eulerian junction structure.

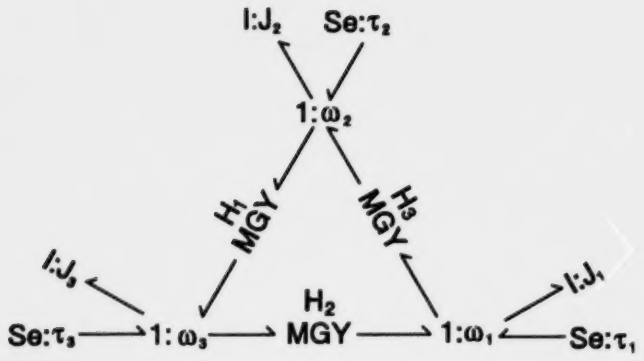


Figure 13: Bond Graph Model of a Euler's Equations

3.4 Multibond Graphs

To deal effectively with vector and tensor quantities, the bond graph notation has been extended to include the concept of multibonds (described in [6, 7, 8]). A multibond represents the grouping of three power bonds, corresponding to an orthogonal coordinate system used to describe the dynamic system. As such, a multibond is not a vector. However, the effort and flow variables associated with the multibond are vectors. For example, in cartesian coordinates, the effort and flow variables become: $\mathbf{e} = \begin{bmatrix} e_x & e_y & e_z \end{bmatrix}^T$ and $\mathbf{f} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T$. The power of a multibond is then given by: $\mathbf{P} = \mathbf{e^T f}$.

Multibond graph methods may be thought of as a generalization of bond graph methods in the same manner that matrix algebra is a generalization of scalar algebra. With this in mind, the translation of bond graph elements into multibond graph elements is a straightforward procedure:

- For passive one-ports, the scalar one-port parameters, R, C, and I, become matrices, and the defining relationships are written in terms of matrix–vector products. To attach a physical meaning, a one-port parameter matrix is referred to as a field parameter. In mechanical systems, the field parameter J is equivalent to a Cartesian tensor of second rank, often called an inertia tensor [15, p. 193].

- Given the power relationship of a multibond, the defining relationships for two-ports must be modified to ensure conservation of power. The new relationships for a transformer are: $e_1 = m^T e_2$ and $m f_1 = f_2$. Similarly, the the relationships for a gyrator element are: $e_1 = m^T f_2$ and $m f_1 = e_2$.

- For three-ports, the defining relationships remain unchanged when translating from bond graphs and multibond graphs except that the scalar sums of the former are replaced with vector sums in the latter.

When dealing with vectors, a new type of relationship needs to be considered: the vector cross product. For ordinary bond graphs, the cross product relationship was introduced using Euler's equations. To extend this to multibonds, equation 3.3 is rewritten as:

$$\dot{\mathbf{H}} = \left(\frac{d\mathbf{H}}{dt}\right)_b + \boldsymbol{\Lambda}\mathbf{H}$$

Here the vector cross product has been replaced with a matrix–vector mul-

tiplication:

$$\boldsymbol{\omega} \times \mathbf{H} = \boldsymbol{\Lambda}\mathbf{H} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix}$$

Recalling that $\mathbf{H} = \mathbf{J}\boldsymbol{\omega}$, the term $\boldsymbol{\Lambda}\mathbf{H}$ can be rewritten as $\boldsymbol{\Lambda}\mathbf{J}\boldsymbol{\omega}$. Since $\boldsymbol{\Lambda}\mathbf{H}$ is an effort, and $\boldsymbol{\omega}$ is a flow, the product $\boldsymbol{\Lambda}\mathbf{J}$ represents a resistance, modulated by the angular velocity. This element is called a gyristor,denoted G.. Using this new element, the Eulerian bond graph (figure 13) can be redrawn using multibond graph notation[3], as shown in figure 14.

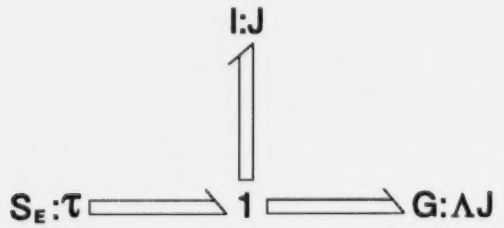


Figure 14: Multibond Graph Model of a Euler's Equations

Applying the time derivative operator (3.1) to the equations $\mathbf{p} = \mathbf{mv}$ and $\mathbf{f} = \dot{\mathbf{p}}$ produces a similar gyristance element for use with linear momentum in an accelerated reference frame.

[3]For aesthetic reasons, the orientation of the arrows is inverted when compared to the notation in [6]

3.5 Link Numbering Scheme and Notation

To create the multibond graph model for a manipulator link in a manner compatible with the N-E method, a link numbering scheme consistent with that used in robotics texts [9] is adopted. In this scheme, the links and joints are numbered from the base outward, with joint (i) connecting link (i) with link $(i-1)$.

An inertial reference frame, numbered 0, is established at the base of the manipulator. At each subsequent joint a coordinate frame, having the same index as the joint number, is attached. Coordinate frames are related to each other through the use of rotation matrices. The notation ${}^{X}R_{Y}$ is used to indicate the rotation from coordinate frame Y to coordinate frame X.

Vector quantities can be manipulated only if they share a common reference frame. Since multiple reference frames are used in robotics, it is necessary to remember which frame a variable is referred to. To keep track of this, a notation very similar to that used for transformations is used. Thus, ${}^{Y}X_{Z}$ is used to indicate the quantity X, belonging to link or joint Z, measured with respect to coordinate frame Y.

3.6 Link Kinematics

The motion of a rigid body, moving in space, can be divided into two parts: the translation of the body CM; and the rotation of the body about its CM. What this means, in terms of developing the multibond graph junction structure, is that the angular velocity of the link can be treated separately

from the linear velocity of the link.

3.6.1 Angular Velocity

Let $\dot{\theta}_i$ represent the angular velocity of link (i), with respect to link $(i-1)$. Then the angular velocity of link (i), ${}^0\omega_i$, with respect to the inertial base coordinate system is given by:

$$ {}^0\omega_i = {}^0\omega_{i-1} + {}^0R_{i-1}\dot{\theta}_i \tag{3.4} $$

In other words, the angular velocity of link (i) , referred to the base, is equal to the angular velocity of link $(i-1)$ referred to the base, plus the angular velocity of link (i) with respect to link $(i-1)$ referred to the base. The multibond junction structure that represents equation 3.4 is shown in figure 15. In this figure, $\dot{\theta}_i$ is shown as a constant flow source, representing a rotary actuator.

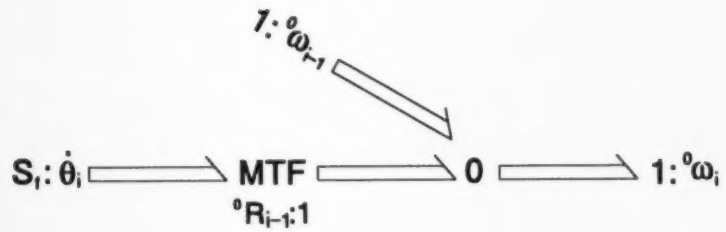


Figure 15: Multibond Graph Junction Structure for Link Angular Velocity

3.6.2 Linear Velocity

Figure 16 shows three coordinate frames, (0),$(i-1)$, and (i) where it is assumed that the $(i-1)$ and (i) frames translate and rotate with respect to

the (0) frame. Referring to the figure, the position vector ${}^0\mathbf{r}_i$ is defined as:

$$ {}^0\mathbf{r}_i = {}^0\mathbf{r}_{i-1} + {}^0\mathbf{r}_i^\star. $$

The velocity is then given by:

$$ {}^0\mathbf{v}_i = \left(\frac{d}{dt}\right)\left({}^0\mathbf{r}_i\right) = \left(\frac{d}{dt}\right)\left({}^0\mathbf{r}_{i-1}\right) + \left(\frac{d}{dt}\right)\left({}^0\mathbf{r}_i^\star\right) \tag{3.5} $$

The first term on the right of this expression is simply the linear velocity of the preceding link:

$$ \left(\frac{d}{dt}\right)\left({}^0\mathbf{r}_{i-1}\right) = {}^0\mathbf{v}_{i-1} \tag{3.6} $$

The second term can be expanded using operator (3.1):

$$ \left(\frac{d}{dt}\right)\left({}^0\mathbf{r}_i^\star\right) = \left(\frac{d}{dt}\right)_{i-1}\left({}^0\mathbf{r}_i^\star\right) + {}^0\boldsymbol{\omega}_{i-1} \times {}^0\mathbf{r}_i^\star \tag{3.7} $$

Letting $\dot{q}_i$ represent the linear velocity of joint (i), with respect to joint $(i-1)$, and once again applying (3.1):

$$ \left(\frac{d}{dt}\right)_{i-1}\left({}^0\mathbf{r}_i^\star\right) = {}^0\mathbf{R}_{i-1}\dot{q}_i + {}^0\mathbf{R}_{i-1}\dot{\theta}_i \times {}^0\mathbf{r}_i^\star \tag{3.8} $$

Combining equation 3.5, with equations 3.6, 3.7 and 3.8 allows the linear velocity of a link joint to be expressed as:

$$ {}^0\mathbf{v}_i = {}^0\mathbf{v}_{i-1} + {}^0\omega_i \times {}^0\mathbf{r}_i^\star + {}^0\mathbf{R}_{i-1}\dot{q}_i \tag{3.9} $$

To obtain the linear velocity of the link CM, the following equation is considered:

$$ {}^0\bar{\mathbf{r}}_i = {}^0\mathbf{r}_i^\star + {}^0\bar{\mathbf{s}}_i $$

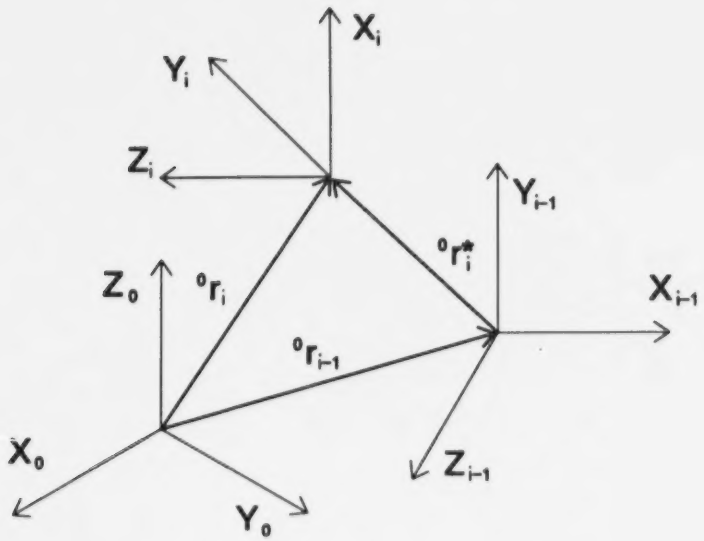


Figure 16: Three Coordinate Frames

Here, ${}^0\bar{\mathbf{r}}_i$ represents a vector from the base inertial coordinate frame; ${}^0\mathbf{r}_i$ is a position vector from the base frame to joint i; and ${}^0\bar{\mathbf{s}}_i$ is a position vector from joint i to the link CM. The velocity of the CM is then given by:

$$ {}^0\bar{\mathbf{v}}_i = {}^0\mathbf{v}_i + \frac{d}{dt}\left({}^0\bar{\mathbf{s}}_i\right) $$

Applying operator (3.1) to the second term on the right yields:

$$ \frac{d}{dt}\left({}^0\bar{\mathbf{s}}_i\right) = \left(\frac{d}{dt}\right)_r \left({}^0\bar{\mathbf{s}}_i\right) + {}^0\boldsymbol{\omega}_i \times {}^0\bar{\mathbf{s}}_i $$

Assuming that the CM is fixed, relative to frame i, the linear velocity of the link CM is:

$$ {}^0\bar{\mathbf{v}}_i = {}^0\mathbf{v}_i + {}^0\boldsymbol{\omega}_i \times {}^0\bar{\mathbf{s}}_i \tag{3.10} $$

3.7 Multibond Graph Link Model

Using expressions 3.4, 3.9, and 3.10, the multibond junction structure for a link can be drawn, as shown in figure 17

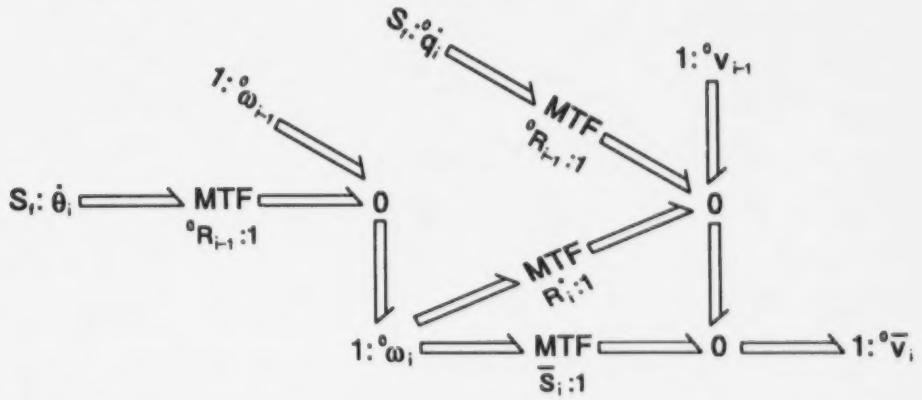


Figure 17: Multibond Graph Link Junction Structure

In this figure, the transformer parameters $\bar{S}_i$ and $R_i^\star$ are given by:

$$\begin{bmatrix} 0 & -{}^0\bar{s}_{i,3} & {}^0\bar{s}_{i,2} \\ {}^0\bar{s}_{i,3} & 0 & -{}^0\bar{s}_{i,1} \\ -{}^0\bar{s}_{i,2} & {}^0\bar{s}_{i,1} & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & -{}^0r^\star_{i,3} & {}^0r^\star_{i,2} \\ {}^0r^\star_{i,3} & 0 & -{}^0r^\star_{i,1} \\ -{}^0r^\star_{i,2} & {}^0r^\star_{i,1} & 0 \end{bmatrix} \tag{3.11}$$

respectively. The numbers $(1, 2, 3)$ indicate the indices of the vectors ${}^0\bar{\mathbf{s}}_i$ and ${}^0\mathbf{r}^\star_i$.

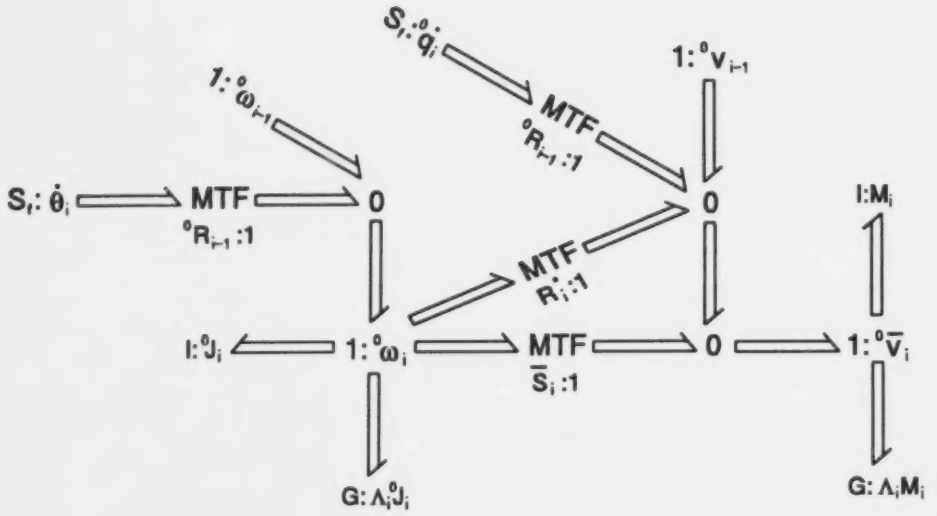


Figure 18: Multibond Graph Link Model

Having obtained the multibond graph link junction structure, it is a simple matter to complete the model by attaching the inertance and gyristance one-ports, introduced in section 3.4, to the structure. The finished model is shown in figure 18. In this figure, the matrix Λ_i is given by:

$$\Lambda_i = \begin{bmatrix} 0 & -{}^0\omega_{i,3} & {}^0\omega_{i,2} \\ {}^0\omega_{i,3} & 0 & -{}^0\omega_{i,1} \\ -{}^0\omega_{i,2} & {}^0\omega_{i,1} & 0 \end{bmatrix}$$

M_i is a scalar representing the link mass, while ${}^0J_i$ is the inertia tensor for the link, with respect to the base inertial reference frame.

In practice, only one of $\dot{\theta}_i$ or $\dot{q}_i$ will be non-zero, depending on whether the link is in rotation or translation. Also, causality considerations may mean that the flow sources representing $\dot{\theta}_i$ or $\dot{q}_i$ have to be replaced with effort

sources.

3.8 Example: Single Link Manipulator

As an example of using the multibond graph link model, the multibond graph for a simple one link rigid manipulator, shown in figure 19 is considered.

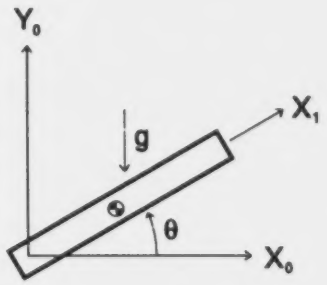


Figure 19: Single-Link Manipulator

For this manipulator, the rotation matrix ${}^0R_1$ is:

$$ {}^0R_1 = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

The parameters ${}^0\boldsymbol{\omega}_0$, ${}^0\mathbf{v}_0$, ${}^0\mathbf{v}_1$, $\mathbf{r}_1^*$, $\dot{q}_1$, and $\Lambda_1\,{}^0J_1$ are all equal to 0. This leads to the simple multibond graph model shown in figure 20.

For this example, the state variable is arbitrarily chosen as the angular momentum of the manipulator, H. This requires that, to satisfy causality, the flow source is replaced with an effort source. An augmented version of the multibond graph is shown in figure 21.

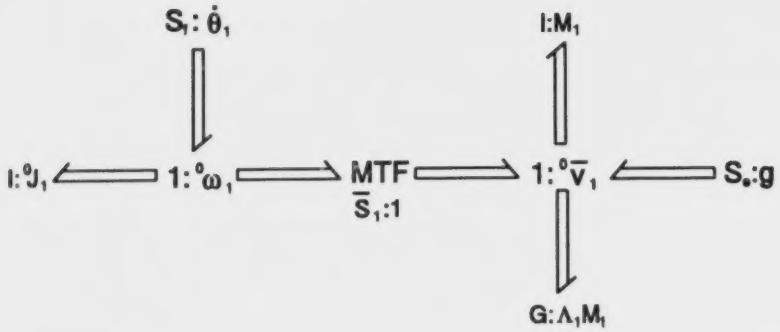


Figure 20: Single-Link Manipulator Multibond Graph

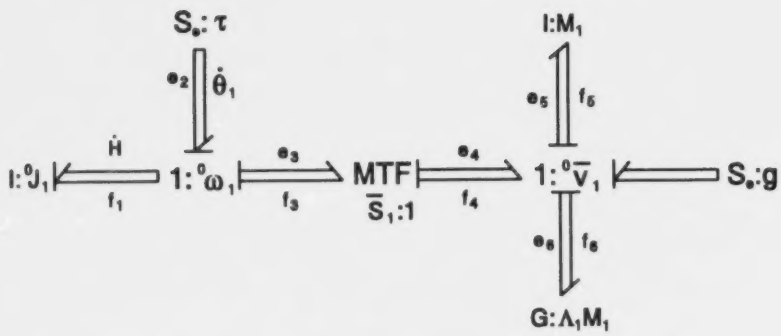


Figure 21: Augmented Single-Link Manipulator Multibond Graph

From the augmented multibond graph, the state equations for the single link manipulator can be determined:

$$f_1 = \left({}^0J_1\right)^{-1} H \tag{3.12}$$

Summing at the 1-junction (${}^0\omega_1$) gives:

$$\dot{H} = e_2 - e_3 \tag{3.13}$$

Assuming that $\tau = \begin{bmatrix} 0 & 0 & \tau \end{bmatrix}^T$:

$$e_2 = \tau = \begin{bmatrix} 0 \\ 0 \\ \tau \end{bmatrix} \tag{3.14}$$

The relationship between e_3 and e_4 is:

$$e_3 = \left(\bar{S}_1\right)^T e_4 \tag{3.15}$$

The gravitational force $Se:g$ acts in the Y_0 direction, and is given by:

$$Se:g = \begin{bmatrix} 0 & -mg & 0 \end{bmatrix}^T \tag{3.16}$$

Summing the efforts at the 1-junction (${}^0\bar{v}_1$) yields:

$$e_4 = e_5 + e_6 - Se:g \tag{3.17}$$

Multibond (5) is in derivative causality so that

$$e_5 = m\frac{d}{dt}(f_5) \tag{3.18}$$

Also,

$$e_6 = \Lambda_1 m f_6 \tag{3.19}$$

The relationship between f_4 and f_3 is:

$$f_4 = \bar{S}_1 f_3 \tag{3.20}$$

Noting that $f_1 = f_2 = f_3$ and $f_4 = f_5 = f_6$, equations 3.12, 3.18, 3.19, and 3.20, can be substituted into 3.17 to yield:

$$e_4 = m\frac{d}{dt}\left(\bar{S}_1 \left({}^0J_1\right)^{-1} H\right) + \Lambda_1 m \bar{S}_1 \left({}^0J_1\right)^{-1} H - Se{:}g \tag{3.21}$$

Then equations 3.14, 3.15 and 3.21 can be combined with equation 3.13 to give:

$$\frac{d}{dt}H = \tau - \left(\bar{S}_1\right)^T \left(m\frac{d}{dt}\left(\bar{S}_1 \left({}^0J_1\right)^{-1} H\right) + \Lambda_1 m \bar{S}_1 \left({}^0J_1\right)^{-1} H - Se{:}g\right) \tag{3.22}$$

With some manipulation this equation could be put into state space form. However, to simplify 3.22, it is easier to substitute ${}^0J_1\,{}^0\omega_1$ for H. This leads to:

$$\frac{d}{dt}\left({}^0J_1\,{}^0\omega_1\right) = \tau - \left(\bar{S}_1\right)^T \left(m\frac{d}{dt}\left(\bar{S}_1\,{}^0\omega_1\right) + m\Lambda_1\bar{S}_1\,{}^0\omega_1 - Se{:}g\right) \tag{3.23}$$

Let the distance to the CM of the link, along X_1, be l. Then

$${}^1\bar{s}_1 = \begin{bmatrix} l & 0 & 0 \end{bmatrix}^T \quad \text{and} \quad {}^0\bar{s}_1 = {}^0R_1\,{}^1\bar{s}_1 = \begin{bmatrix} lcos(\theta) \\ lsin(\theta) \\ 0 \end{bmatrix}$$

From this, using 3.11, $\bar{S}_1$ can be determined:

$$\bar{S}_1 = \begin{bmatrix} 0 & 0 & -lsin(\theta) \\ 0 & 0 & lcos(\theta) \\ lsin(\theta) & -lcos(\theta) & 0 \end{bmatrix} \tag{3.24}$$

At the CM, the inertia matrix is:

$$ {}^1J_1 = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} $$

Referring this to the base:

$$ {}^0J_1 = {}^0R_1 \, {}^1J_1 = \begin{bmatrix} J_{xx}cos\theta & -J_{yy}sin(\theta) & 0 \\ J_{xx}sin(\theta) & J_{yy}cos(\theta) & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} $$

Since the angular velocity of the link, ${}^0\omega_1$ is given by $\begin{bmatrix} 0 & 0 & \dot{\theta} \end{bmatrix}^T$, the derivative of ${}^0J_1 \, {}^0\omega_1$ is:

$$ \frac{d}{dt}\left({}^0J_1 \, {}^0\omega_1\right) = \begin{bmatrix} 0 \\ 0 \\ J_{zz}\ddot{\theta} \end{bmatrix} \tag{3.25} $$

The product $\bar{S}_1 \, {}^0\omega_1$ is:

$$ \bar{S}_1 \, {}^0\omega_1 = \begin{bmatrix} 0 & 0 & -lsin(\theta) \\ 0 & 0 & lcos(\theta) \\ lsin(\theta)\dot{\theta} & -lcos(\theta)\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -lsin(\theta)\dot{\theta} \\ lcos(\theta)\dot{\theta} \\ 0 \end{bmatrix} \tag{3.26} $$

The time derivative of 3.26 is:

$$ \frac{d}{dt}\left(\bar{S}_1 \, {}^0\omega_1\right) = \begin{bmatrix} -lcos(\theta)\dot{\theta}^2 - lsin(\theta)\ddot{\theta} \\ lcos(\theta)\ddot{\theta} - lsin(\theta)\dot{\theta}^2 \\ 0 \end{bmatrix} \tag{3.27} $$

Also using 3.26, the product $\Lambda_1 \bar{S}_1 {}^0\omega_1$ is:

$$\Lambda_1 \bar{S}_1 {}^0\omega_1 = \begin{bmatrix} 0 & -\dot{\theta} & 0 \\ \dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -lsin(\theta)\dot{\theta} \\ lcos(\theta)\dot{\theta} \\ 0 \end{bmatrix} = \begin{bmatrix} -lcos(\theta)\dot{\theta}^2 \\ -lsin(\theta)\dot{\theta}^2 \\ 0 \end{bmatrix} \tag{3.28}$$

Using these results, 3.23 can be rewritten as:

$$\begin{bmatrix} 0 \\ 0 \\ J_{zz}\ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau \end{bmatrix} - m\left(\bar{S}_1\right)^T \begin{bmatrix} -lsin(\theta)\ddot{\theta} - 2lcos(\theta)\dot{\theta}^2 \\ lcos(\theta)\ddot{\theta} - 2lsin(\theta)\dot{\theta}^2 + g \\ 0 \end{bmatrix} \tag{3.29}$$

Finally, this can be simplified to:

$$\begin{bmatrix} 0 \\ 0 \\ J_{zz}\ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau \end{bmatrix} - ml^2 \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta} + \frac{g}{l}cos(\theta) \end{bmatrix} \tag{3.30}$$

Setting the input torque to 0 and dropping the subscript on J, the z component of equation 3.30 is readily recognized as being the equation for a physical pendulum:

$$\ddot{\theta} + \frac{mglcos\theta}{J + ml^2} = 0$$

3.9 Obtaining the N-E Equations From the Multibond Graph Link Model

As mentioned previously, the approach taken to developing the multibond graph link model is very similar to that used by the N-E method. In fact,

with a suitable causal assignment, the N-E equations can be obtained from the multibond graph link model. This provides a convenient mechanism for verifying the correctness of the multibond graph model. In addition, it serves to illustrate the flexibility of the multibond graph approach; the causal assignment that leads to the N-E equations is only one of the several possible assignments.

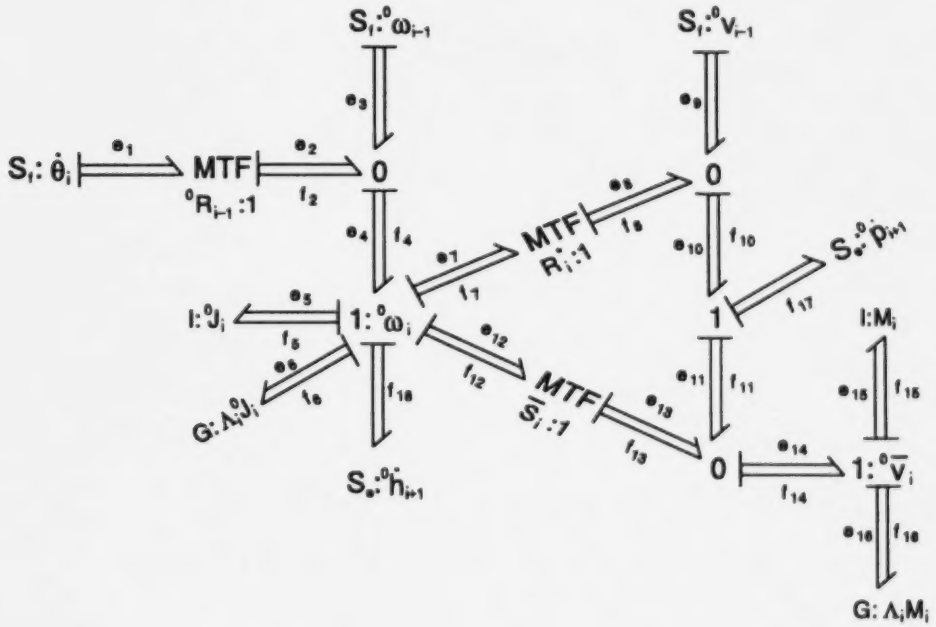


Figure 22: Newton-Euler Multibond Graph Link Model

To obtain the N-E equations for a rotating link, the multibond graph link model is redrawn and augmented as shown in figure 22. In this figure, the connections to link $(i-1)$ are made through constant flow sources. In a multi-link model, these connections would be replaced by 1-junctions (that

also have a common flow). Similarly, the connections to link $(i+1)$ are made through constant effort sources that would be replaced by 0-junctions in a multi-link chain. Taken together, these connections graphically represent the fact that velocities propagate forward along the kinematic chain, while forces propagate backwards.

Another important feature of these connections is that they, in conjunction with the constant flow source representing $\dot{\theta}_i$, place the inertance one-ports in derivative causality. Thus, as with the N-E method, the angular and linear accelerations of the link must be determined before equations can be written for the link.

Since the multibond graph link junction structure was developed using the same velocity relationships used by the N-E approach, the link velocity equations (3.4, 3.9, and 3.10) must be obtainable by summation at the 0-junctions of the multibond graph.

The angular acceleration of the link can be obtained by differentiating equation 3.4:

$$\left(\frac{d}{dt}\right)_s \left({}^0\omega_i\right) = \left(\frac{d}{dt}\right)_s \left({}^0\omega_{i-1}\right) + \left(\frac{d}{dt}\right)_s \left({}^0R_{i-1}\,\dot{\theta}_i\right)$$

Applying operator (3.1) to the second term on the right gives:

$$\left(\frac{d}{dt}\right)_s \left({}^0R_{i-1}\,\dot{\theta}_i\right) = \left(\frac{d}{dt}\right)_{i-1} \left({}^0R_{i-1}\,\dot{\theta}_i\right) + {}^0\omega_{i-1} \times \left({}^0R_{i-1}\,\dot{\theta}_i\right)$$

Also:

$$\left(\frac{d}{dt}\right)_{i-1} \left({}^0R_{i-1}\,\dot{\theta}_i\right) = {}^0R_{i-1}\,\ddot{\theta}_i$$

Using these results, the angular acceleration of the link at joint (i) can be

expressed as:

$$ {}^0\dot{\omega}_i = {}^0\dot{\omega}_{i-1} + {}^0R_{i-1}\,\ddot{\theta}_i + {}^0\omega_{i-1} \times \left({}^0R_{i-1}\,\dot{\theta}_i\right) \tag{3.31} $$

Similarly, the linear acceleration of the link at joint (i) can be obtained by differentiating 3.9:

$$ \left(\frac{d}{dt}\right)_s \left({}^0v_i\right) = \left(\frac{d}{dt}\right)_s \left({}^0v_{i-1}\right) + \left(\frac{d}{dt}\right)_s \left({}^0\omega_i \times {}^0r_i^\star\right) $$

Here the term involving $\dot{q}_i$ has been dropped because the link is not translational. Once again applying (3.1) to this equation, the acceleration at the joint can be written as:

$$ {}^0\dot{v}_i = {}^0\dot{v}_{i-1} + {}^0\dot{\omega}_i \times {}^0r_i^\star + {}^0\omega_i \times \left({}^0\omega_i \times {}^0r_i^\star\right) \tag{3.32} $$

The linear acceleration at the link CM can be obtained by differentiating in the same manner as was done to obtain the linear and angular accelerations at the joint. The resulting acceleration at the link CM is:

$$ {}^0\bar{a}_i = {}^0\dot{\omega}_i \times {}^0\bar{s}_i + {}^0\omega_i \times \left({}^0\omega_i \times {}^0\bar{s}_i\right) + {}^0\dot{v}_i \tag{3.33} $$

Expressions for the torques and forces acting on the link are obtained by summing at the 1-junctions of figure 22. At the junction (${}^0\bar{v}_i$):

$$ e_{14} = e_{15} + e_{16} \tag{3.34} $$

$$ e_{15} = M_i \dot{f}_{15} $$

$$ e_{16} = \Lambda_i M_i f_{16} $$

The term $\dot{f}_{15}$ represents the derivative of ${}^0\bar{v}_i$ with respect to frame (i). Since the link is rigid this is the same as the derivative, with respect to frame (i), of ${}^0v_i$:

$$e_{15} = \left(\frac{d}{dt}\right)_i \left({}^0\bar{v}_i\right) = \left(\frac{d}{dt}\right)_i \left({}^0v_i\right)$$

Also, the term $\Lambda_i M_i f_{16}$ can be rewritten as the vector cross product:

$$e_{16} = M_i \left({}^0\omega_i \times {}^0\,\bar{v}_i\right)$$

Substituting 3.10 into this equation, and noting that:

$$\left(\frac{d}{dt}\right)_i \left({}^0v_i\right) + {}^0\omega_i \times {}^0v_i = {}^0\dot{v}_i$$

Equation 3.34 can be written as:

$$e_{14} = M_i \left({}^0\dot{\omega}_i \times {}^0\bar{s}_i + {}^0\omega_i \times \left({}^0\omega_i \times {}^0\bar{s}_i\right) + {}^0\dot{v}_i\right)$$

The term in the outer parentheses is simply the linear acceleration of the link CM, so that this can be rewritten as:

$$F_i = M_i\,{}^0\bar{a}_i \tag{3.35}$$

Here the effort e_{14} has been relabeled F_i.

Summation at the 1-junction representing the linear velocity of the joint (not labeled in figure 22) yields:

$$e_{10} = F_i + {}^0\dot{p}_{i+1} \tag{3.36}$$

The torque acting on the link is given by:

$$e_4 = {}^0\dot{h}_{i+1} + (e_5 + e_6) + (e_7 + e_{12}) \tag{3.37}$$

The term $(e_5 + e_6)$ can be relabeled N_i and is given by:

$$N_i = {}^0J_i\,{}^0\dot{\omega}_i + {}^0\omega_i \times \left({}^0J_i\,{}^0\omega_i\right) \tag{3.38}$$

The matrices $\bar{S}_i$ and $R_i^\star$ are anti-symmetric, so that $(\bar{S}_i)^T = -\bar{S}_i$ and $(R_i^\star)^T = -R_i^\star$. The sum of e_7 and e_{12} can then be given as:

$$e_7 + e_{12} = {}^0r_i^\star \times {}^0\dot{p}_{i+1} + \left({}^0r_i^\star + {}^0\bar{s}_i\right) \times F_i$$

Then, equation 3.37 can be rewritten as:

$$e_4 = {}^0\dot{h}_{i+1} + N_i + {}^0r_i^\star \times {}^0\dot{p}_{i+1} + \left({}^0r_i^\star + {}^0\bar{s}_i\right) \times F_i \tag{3.39}$$

Finally, the input torque to the link is given from the multibond graph as:

$$\tau_i = \left({}^0R_{i-1}\right)^T e_4 \tag{3.40}$$

Equations 3.4, 3.31, 3.32, 3.33, 3.35, 3.36, 3.38, 3.39, and 3.40 constitute the Newton-Euler equations for a rotational link [14, p. 114], and are repeated below for convenience.

- Forward Equations:

$${}^0\omega_i = {}^0\omega_{i-1} + {}^0R_{i-1}\dot{\theta}_i$$

$${}^0\dot{\omega}_i = {}^0\dot{\omega}_{i-1} + {}^0R_{i-1}\,\ddot{\theta}_i + {}^0\omega_{i-1} \times \left({}^0R_{i-1}\,\dot{\theta}_i\right)$$

$${}^0\dot{v}_i = {}^0\dot{v}_{i-1} + {}^0\dot{\omega}_i \times {}^0r_i^\star + {}^0\omega_i \times \left({}^0\omega_i \times {}^0r_i^\star\right)$$

$${}^0\bar{a}_i = {}^0\dot{\omega}_i \times {}^0\bar{s}_i + {}^0\omega_i \times \left({}^0\omega_i \times {}^0\bar{s}_i\right) + {}^0\dot{v}_i$$

- Backwards Equations:

$$F_i = M_i {}^0\bar{a}_i$$

$$N_i = {}^0J_i {}^0\dot{\omega}_i + {}^0\omega_i \times \left({}^0J_i {}^0\omega_i\right)$$

$$e_{10} = F_i + {}^0\dot{p}_{i+1}$$

$$e_4 = {}^0\dot{h}_{i+1} + N_i + {}^0r_i^\star \times {}^0\dot{p}_{i+1} + \left({}^0r_i^\star + {}^0\bar{s}_i\right) \times F_i$$

$$\tau_i = \left({}^0R_{i-1}\right)^T e_4$$

3.10 Summary

In this chapter the bond graph concept was extended to include multibond graphs. Then, using multibond graph elements, a model for a single manipulator link was developed. This link model is suitable for use as a component to construct arbitrary manipulator configurations.

The correctness of the multibond graph link model was demonstrated by deriving the N-E equations from an augmented version of the graph. Interestingly, the need to use forward and backward iteration while solving the N-E equations, is readily visible in the multibond graph model.

A simple example (section 3.8) demonstrated the extraction of equations from the multibond graph model of a manipulator. Even for this one-link case, reducing the equations to a simple form is tedious. In the next chapter, the simulation of multibond graph models is presented. Combining computer simulation with the multibond graph model of a manipulator, the dynamics can be studied without obtaining explicit closed-form equations.

Chapter 4

Simulating Multibond Graphs with SIMULINK

4.1 Introduction

Because of the importance of simulation in the study of multi-body dynamic systems,such as robots, extensive research has been carried out to develop efficient formulations of the dynamic equations [9, pp. 216-218] [2, p. 155]. Interestingly, the emphasis on efficient simulation algorithms applies only to the execution time of the simulation, not the time taken to construct the dynamic model.

In contrast, Beauwin and Lorenz present a methodology whereby bond graph models of subsystems are connected to construct complicated mechanisms [3]. The difference between this approach to simulation, and those that consider efficiency, is very similar to the difference between interpreted and

compiled languages within the field of computer programming. Interpreted languages encourage interaction and exploration, while compiled languages emphasize speed of execution.

The approach presented in [3] complements the one presented in this thesis: Beauwin and Lorenz model the mechanism primarily as a collection of joints; the approach taken here is to model the system as a collection of links. This difference is largely due to the fact that conventional approaches to developing the kinematics and dynamics for robotic systems are based on the use of links.

In this chapter, a procedure which will enable us to use the multibond graph model of a robotic manipulator to simulate its dynamic response is described. This can be done without obtaining a closed-form solution of the dynamic equations of the robotic manipulator.

There are a number of different ways to simulate the dynamics of a robotic system using the bond graph model. If numerical efficiency is the prime requisite, iterative dynamic equations, such as the N-E equations developed in chapter 3, can be obtained and simulated using a compiled programming language. On the other hand, if some of the execution speed can be sacrificed, simulation can be undertaken using a more flexible and interactive software environment.

An important characteristic of interactive simulation environments is that the simulation can be presented graphically. Since bond graph modeling techniques derive much of their strength from the highly visual manner with which the system dynamics are presented, it seems appropriate that the graphical representation be retained when simulating the system. This can

be accomplished in two different ways:

1. Use software written expressly for simulating multibond graphs.

2. Transform the multibond graph model into a form suitable for schematic input into a block-diagram modeling software package such as VisSim or SIMULINK.

While the first alternative provides an easy and direct approach to simulate multibond graphs, it lacks flexibility when compared to the second alternative. Block-diagram modeling software packages include the capability to simulate multibond graphs as one feature of a much larger package. For example, most block-diagram modeling software packages allow mixed continuous and discrete simulations to be created. This makes it possible to simulate a digital controller attached to the multibond graph model of a manipulator, as well as the multibond graph model itself.

Of the many block-diagram modeling software packages available, SIMULINK, in conjunction with its underlying simulation engine MATLAB, appears to be the most suitable for simulating multibond graphs. The reason for this is that SIMULINK allows the passing of vectors for block inputs and outputs. VisSim and Tutsim, two other packages which can be used, lack the ability to handle vectors and require that the multibond graph be decomposed into an ordinary bond graph.[7] Noting that flexibility and interaction are properties of the simulation software, and identifying these properties as goals, the simulation of multibond graph models of robotic manipulators using SIMULINK is presented here.

4.2 Simulation and Derivative Causality

A potential obstacle to the simulation of a multibond graph is the presence of derivative causality in the augmented graph. This indicates that the graph contains algebraic loops called "class 1 zero-order causal paths."[10] In order to simulate bond graphs containing these paths it is necessary to either: eliminate the zero-order path using a variety of techniques; or acceept the zero-order path and simulate the multibond graph using algorithms developed for solving stiff systems.

Elimination of zero-order causal paths can be done analytically, through transference of dependent storage elements to those with integral causality. Dijk [11] citing [5] indicates that, for large complex systems, this may not always be possible. An alternative approach involves relaxing the derivative constraint through the introduction of stiff compliances with, or without, parasitic resistances into the model. Finally, it is possible to insert source elements into the bond graph that model the constraint forces with Lagrange multipliers.

The price paid for accepting derivative causality, and using stiff-equation solvers, is increased simulation time. Still, Dijk contends that the increased accuracy obtainable with this approach makes the acceptance of derivative causality preferable to elimination [11].

4.3 A SIMULINK MTF Block

Simulation of multibond graph models using block-diagram modeling software is possible because each multi-port is defined in terms of a mathematical input-output relationship. SIMULINK contains built-in blocks that represent all the mathematical relationships to be found in a multibond graph, with the exception of the modulated transformer (MTF). Before simulating multibond graphs using SIMULINK, then, it is necessary to create a SIMULINK block corresponding to the MTF.

Information for drawing and simulating a block diagram in SIMULINK is contained within a special MATLAB function called an S-function. This function, like a MATLAB function, can be either an ascii text file called an M-file, or a compiled C or Fortran file called a MEX-file.[18, p. 3-1] While an M-file version of an S-function can be directly entered into a text file, it is more typically generated using mouse-driven commands within a SIMULINK *block diagram* window. Once an S-function has been created, a masking function can be applied to it in order to specify the appearance of the block, and any variable parameters that are to be passed into the block. [18, p. 3-19]

S-functions are comprised of a collection of one or more primitive functions. The majority of these primitive functions are intrinsic to SIMULINK. However, for those instances where a suitable transfer function cannot be constructed using intrinsic functions, provision has been made to construct user-defined S-functions. By definition, intrinsic and user-defined functions are also S-functions.

For the purposes of simulating a multibond graph with SIMULINK, an MTF represents a matrix-vector multiplication, where the matrix elements are functions of one or more time-dependent modulation parameters. Since MATLAB does not allow matrices to contain uninitialized variables, it is necessary to generate the matrix at each time step of the simulation. One way that this can be accomplished is to place the MATLAB commands needed to generate the matrix in a text string, and then pass this string to the MATLAB *eval* function. The *eval* function accepts a text string as input, and evaluates the string as a MATLAB command. This approach to generating the modulation matrix can be expected to slow the simulation significantly, but it offers the advantage that modulation matrix parameters can be entered in the SIMULINK block diagram window via the masking function.

User-defined S-functions within SIMULINK have single input and output vectors (connections). For the MTF block, however, it is desirable to have a separate input for the modulation parameters. Also, for convenience, the number of modulation parameters should be passed into the S-function using the masking function. This means that the MTF block must be contained in two S-functions. The first S-function, constructed using blocks built into SIMULINK, is not limited to having a single input and output vector. As shown in figure 23, this S-function multiplexes the signal input vector with the vector of modulation parameters. The resulting vector is the passed, along with a text string containing the MATLAB commands to generate the modulation matrix, to a user-defined S-function written as a MATLAB M-file. The resulting MTF block, consisting of both S-functions, is easy to use and completely configurable (see figure 24) from the SIMULINK block

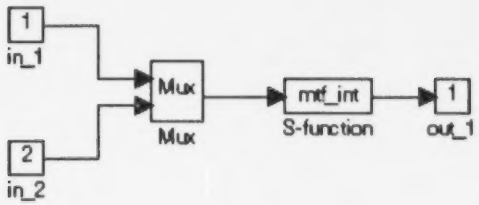


Figure 23: First part of MTF S-function

diagram window. The S-functions used to construct this block are shown in appendix A.

4.4 A SIMULINK Link Model

In chapter 3 it was shown how the multibond graph model for any rigid-body serial manipulator can be constructed using a single component multibond graph. A similar approach is used to construct a SIMULINK model, from the component multibond graph, that is then used as a building block to simulate the dynamics of an arbitrary rigid-body serial manipulator.

Figure 25 shows an augmented version of the link multibond graph model where causality is assigned so that the inertance one-ports, associated with the link angular velocity, are in integral causality. Conversion of this model into a SIMULINK block-diagram model is a simple procedure that involves writing down the multiport relationships from the multibond graph, and implementing the relationships in the block diagram. For example, the relationships at the 1-junction (${}^0\omega_i$) are:

$$e_5 = e_4 - e_6 - e_7 - e_{12} - {}^0N_{i+1}$$

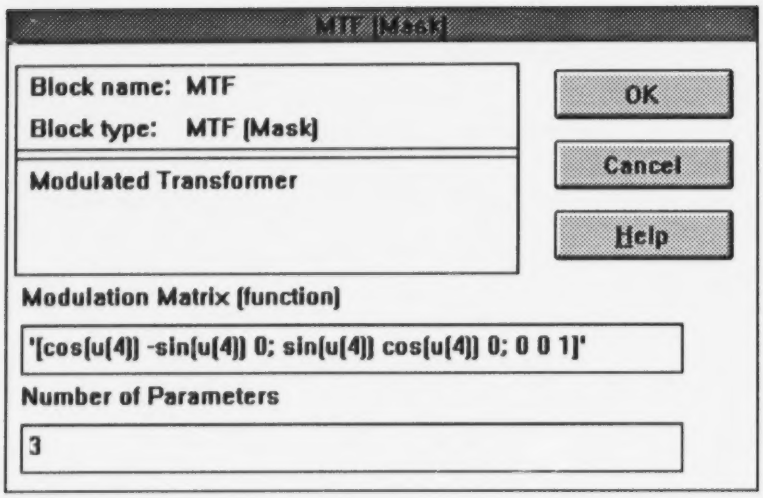


Figure 24: Dialog for entering MTF parameters

and

$$f_5 = f_5 = f_6 = f_7 = f_{12}$$

This is represented as a summing block having five inputs and a single output. Also from the multibond graph:

$$f_5 = ({}^0J_1)^{-1} \int e_5$$

and

$$e_6 = \Lambda_i {}^0J_i f_6$$

The portion of the block diagram corresponding to these equations is shown in figure 26. The complete block diagram for the link, incorporating all of the equations obtained from the corresponding augmented multibond graph (figure 25) is shown in figure 27. The equations are listed in B.

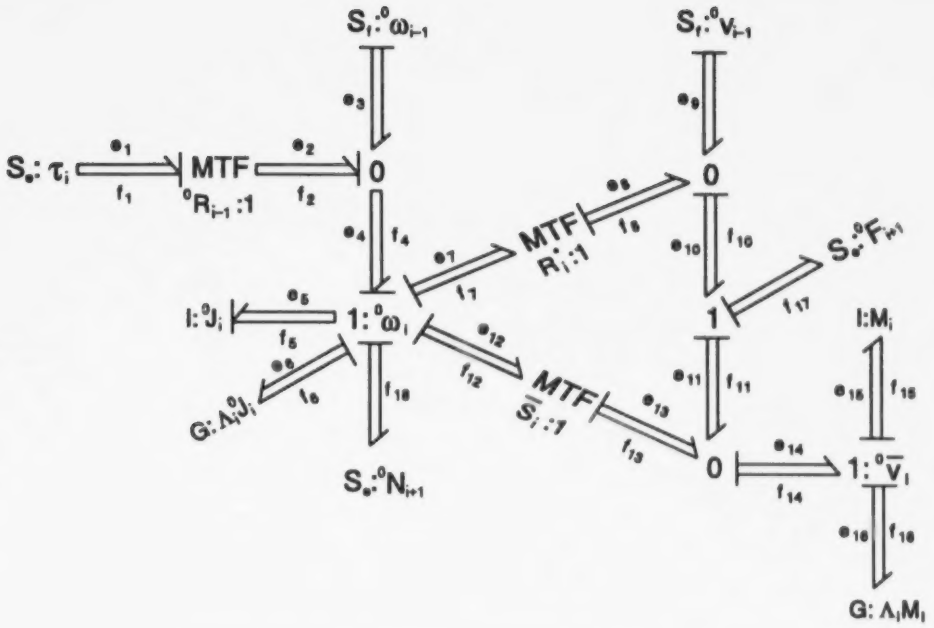


Figure 25: Augmented Multibond Graph Model

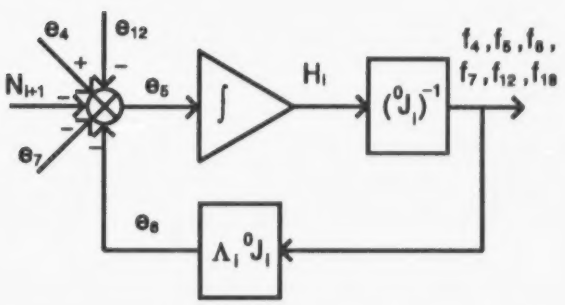


Figure 26: A Part of the Link Block Diagram

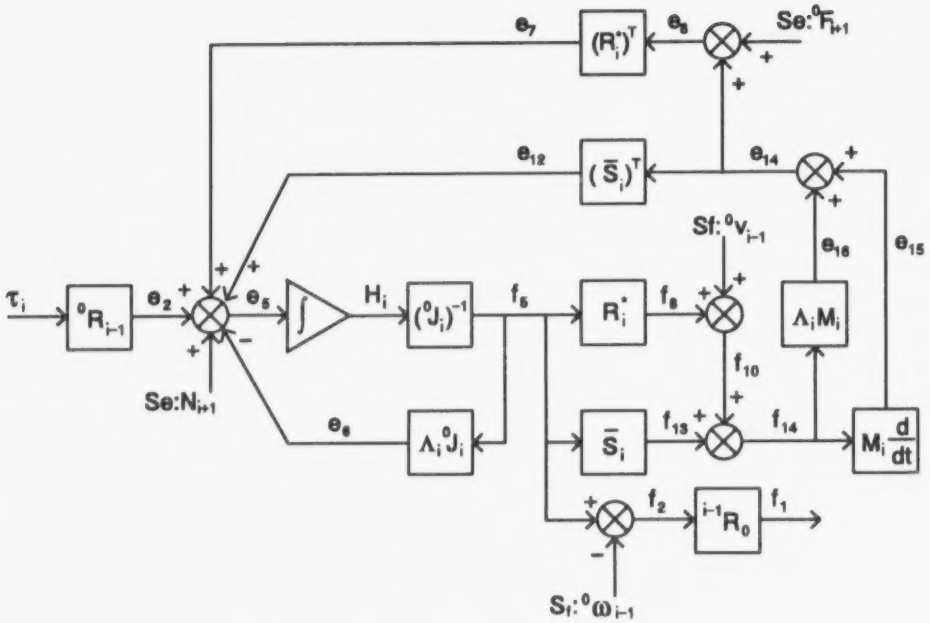


Figure 27: Single Link Block Diagram

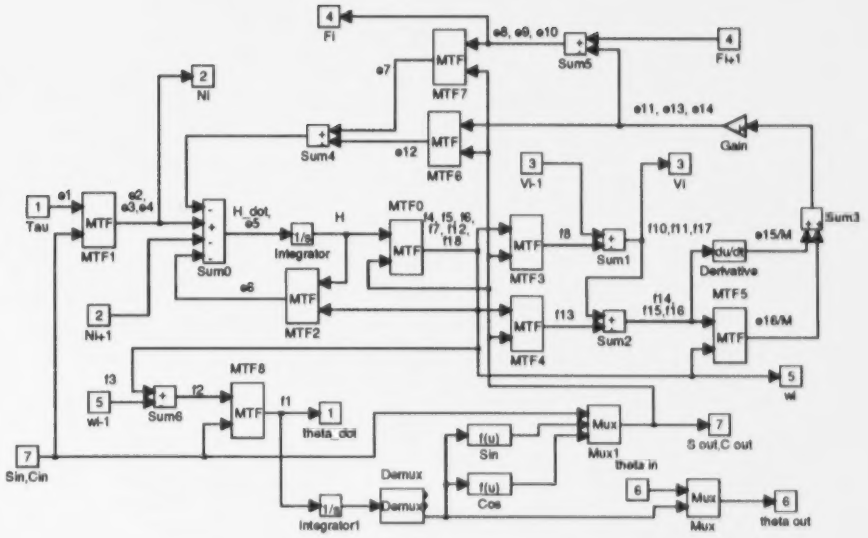


Figure 28: SIMULINK Link Block Diagram

A comparison of this block diagram with the multibond graph model (figure 25) shows that, in appearance, the block-diagram model is at least as simple as the multibond graph model. However, whereas the link forces and velocities are tightly grouped in the multibond graph model they are scattered all over the block diagram. This has important implications if changes to the model are contemplated. For instance, suppose the model needs to be altered to include the effects of joint compliance. For the multibond graph, the change involves attaching a capacitance one-port to the appropriate junction. Modifying the block diagram model is not as obvious or straightforward.

Figure 29: SIMULINK Link Component Block

The SIMULINK block-diagram model shown in figure 28 is considerably more complicated than the model shown in figure 27. One reason for this is that signal lines for the MTF parameters have been added explicitly to the SIMULINK model. A secondary reason is that additional blocks have been added to precalculate the values for $cos(\theta_i)$ and $sin(\theta_i)$. This is done to avoid repeated calls to these functions at each iteration of the simulation.

The functionality of the SIMULINK link model can be encapsulated in a single block, as shown in figure 29, where only the external connections to the link are visible. The inputs and outputs to this block are numbered sequentially from the top down, to correspond with the input and output blocks numbered 1 through 7 in figure 28. The compact version of the link model (figure 29) can be used as a component to construct models for arbitrary rigid-body serial manipulators with revolute joints.

4.5 Example: Two-link Planar Manipulator

As an example of generating a SIMULINK model using the link component block, the model of a two-link planar manipulator, shown in figure 30, is considered. Each link is represented by a separate SIMULINK link compo-

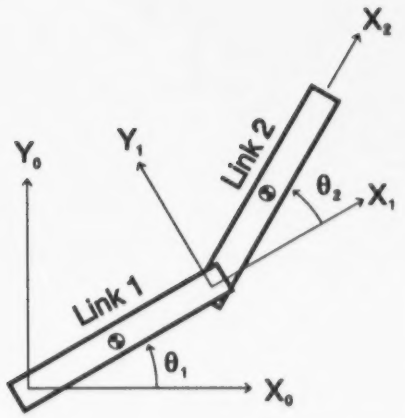


Figure 30: Two-link Planar Manipulator

nent block. The blocks are then connected according to the multibond graph model of figure 25, i.e. velocities propagate forward along the kinematic chain, and forces propagate backwards. Following this, the MTF parameters for each link are entered using the kinematic configuration of figure 30. These procedures are described in the following section.

4.5.1 SIMULINK Model Construction

Model construction starts by *dragging* two copies of the link component block into an empty SIMULINK block-diagram window. Once the blocks are in place, connections between them can be established:

1. Output 2 (N_i) of link 2 is connected to input 2 (N_{i+1}) of link 1.
2. Output 3 (v_i) of link 1 is connected to input 3 (v_{i-1}) of link 2.
3. Output 4 (F_{i+1}) of link 2 is connected to input 4 (F_i) of link 1.

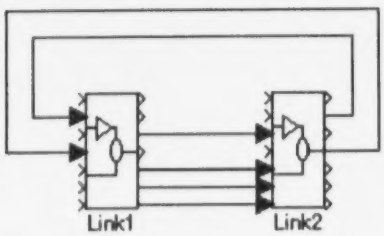


Figure 31: First Stage of Model Construction

4. Output 5 (ω_i) of link 1 is connected to input 5 (ω_{i-1}) of link 2.

5. Output 6 (theta out) of link 1 is connected to input 6 (theta in) of link 2.

6. Output 7 (S out, C out) of link 1 is connected to input 7 (S in, C in) of link 2.

A picture of the model, at this stage of construction, is shown in figure 31.

With the two links connected, the MTF parameters for each link can be entered. The rotation matrices for the manipulator are:

$$ {}^0R_1 = \begin{bmatrix} cos(\theta_1) & -sin(\theta_1) & 0 \\ sin(\theta_1) & cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } {}^1R_2 = \begin{bmatrix} cos(\theta_2) & -sin(\theta_2) & 0 \\ sin(\theta_2) & cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

For this example, each of the links is assumed to be rectangular, with a length of 1 meter, and a cross-section of $0.1m. \times 0.1m.$. The first link has a mass of 60 kilograms, while the second link has a mass of 30 kilograms. Using these

values, the inertia tensors for the links are calculated to be [13, p 227]:

$$ {}^1J_1 = \begin{bmatrix} 0.10 & 0 & 0 \\ 0 & 5.0 & 0 \\ 0 & 0 & 5.0 \end{bmatrix} \quad \text{and} \quad {}^2J_2 = \begin{bmatrix} 0.05 & 0 & 0 \\ 0 & 2.5 & 0 \\ 0 & 0 & 2.5 \end{bmatrix} $$

Also the vectors ${}^0\mathbf{r}_1^\star$, ${}^0\mathbf{r}_2^\star$, ${}^0\bar{\mathbf{s}}_1$, and ${}^0\bar{\mathbf{s}}_2$ are given by:

$$ {}^0\mathbf{r}_1^\star = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \; {}^0\mathbf{r}_2^\star = \begin{bmatrix} cos(\theta_1) \\ sin(\theta_1) \\ 0 \end{bmatrix}, \; {}^0\bar{\mathbf{s}}_1 = \begin{bmatrix} 0.5cos(\theta_1) \\ 0.5sin(\theta_1) \\ 0 \end{bmatrix} $$

and

$$ {}^0\bar{\mathbf{s}}_2 = \begin{bmatrix} 0.5(cos(\theta_1)cos(\theta_2) - sin(\theta_1)sin(\theta_2)) \\ 0.5(sin(\theta_1)cos(\theta_2) + sin(\theta_2)cos(\theta_1)) \\ 0 \end{bmatrix} $$

With reference to figures 27 and 28, the MTF parameters for link 1 are as follows:

1. The block labeled MTF0 is the matrix:

$$ ({}^0J_1)^{-1} = \begin{bmatrix} 10cos(\theta_1) & 10sin(\theta_1) & 0 \\ -0.2sin(\theta_1) & 0.2cos(\theta_1) & 0 \\ 0 & 0 & 0.2 \end{bmatrix} $$

   This block has two modulation parameters, $sin(\theta_1)$ and $cos(\theta_1)$.

2. The block labeled MTF1 represents the rotation ${}^0R_0$, which is the identity matrix. To avoid unnecesary calculations, this block is removed. Similarly, MTF8 is also removed.

3. Block MTF2 represents the cross-product $\omega \times H$. Because link 1 is not in translation, this block can be removed as well.

4. Blocks MTF3 and MTF7 represent the matrix R_1^* and its transpose respectively. For link 1 these are both empty matrices, but are left in the model for simplicity.

5. Blocks MTF4 and MTF6 represent the matrix $\bar{S}_1$ and it transpose. These blocks are modulated by the parameters $sin(\theta_1)$ and $cos(\theta_1)$.

6. Finally, block MTF5 represents the matrix Λ_1, with its three modulation parameters: $\omega_{1,1}$, $\omega_{1,2}$, and $\omega_{1,3}$.

As mentioned previously, $sin(\theta_1)$ and $cos(\theta_1)$ are pre-calculated in the block. When entered as modulation parameters, these two values are made accessible to the user-defined MTF S-function as $u(4)$ and $u(5)$, as dictated by SIMULINK. Thus, the rotation matrix ${}^0R_1$ is entered in the appropriate MTF dialog box as:

```
"[u(5) -u(4) 0; u(4) u(5) 0; 0 0 1]"
```

With the exception that no blocks can be eliminated, entry of the MTF parameters for link 2 follows in a similar manner to that of link 1. The variables $sin(\theta_2)$ and $cos(\theta_2)$ are available as MTF parameters $u(6)$ and $u(7)$; the parameters $u(4)$ and $u(5)$ are also available, having been passed into the link.

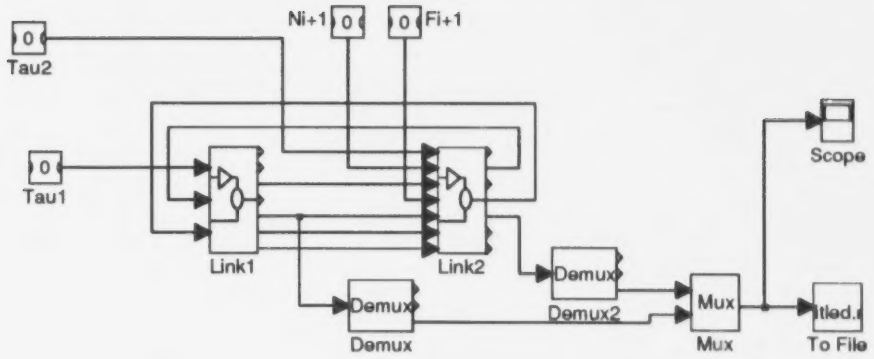


Figure 32: SIMULINK Model of Two-link Planar Manipulator

4.5.2 Simulation Results

To test the SIMULINK model, external connections are established as shown in figure 32. To simulate this model, both torque inputs are set to 0, and initial conditions are set so that $H_1(0) = 0.5$, and $H_2(0) = 1$. The system is simulated using Gear integration. This is necessary because both links contain derivative blocks. To obtain accurate results, the simulation *tolerance* must be set at `1e-7` or smaller.

As well as setting the tolerance of the simulation, the maximum and minimum step sizes must also be set. To determine the optimum settings for these parameters, a series of short simulations were run, varying these parameters. The results are shown in table 4.5. Interestingly, a minimum step size of 10^{-2} seconds produces surprisingly long simulation times.

Simulation results, showing the angular velocities of the links, are shown in figure 33. From the plot it is seen that, since the angular momentum of

Max. Step (S)	Min. Step (S)						
	1e-1	1e-2	1e-3	1e-4	1e-5	1e-6	1e-7
1	4.97e+2	2.85e+3	1.44e+3	1.35e+3	1.42e+3	1.35e+3	1.37e+3
1e-1	5.01e+2	2.86e+3	1.45e+3	1.35e+3	1.42e+3	1.35e+3	1.37e+3
1e-2	5.01e+2	2.86e+3	1.39e+3	1.42e+3	1.46e+3	1.39e+3	1.42e+3
1e-3	5.02e+2	2.86e+3	4.07e+2	4.10e+2	4.11e+2	4.10e+2	4.15e+2
1e-4	5.04e+2	2.86e+3	4.05e+2	4.12e+3	4.12e+3	4.15e+3	4.10e+3

Table 4.5: Results of Varying Maximum and Minimum Step Sizes

link 1 is constant, the angular velocity of the link is constant. The angular velocity of link 2 is periodic and discontinuous; the discontinuity results in simulation time of just over one hour, using a maximum step size of 10^{-3} seconds and minimum step size of 10^{-4} seconds.

This simulation result is somewhat counter-intuitive. If the two links were freely rotating it would be expected that a torque, exerted by the second link on the first, would influence the angular velocity of the first link. But, from the causality assignment of the augmented multibond graph model (25), it can be seen that the velocities ${}^0\omega_i$ and ${}^0v_i$ propagate forward along the manipulator chain. Setting the input torques to zero, then, is equivalent to driving the links with constant velocity sources. That is, ${}^0\theta_1$ and ${}^0\theta_2$ are constant. With this in mind, the simulation results are as expected.

All simulations were run on an HP 735 workstation; the S-function containing the two-link manipulator is shown in appendix C.

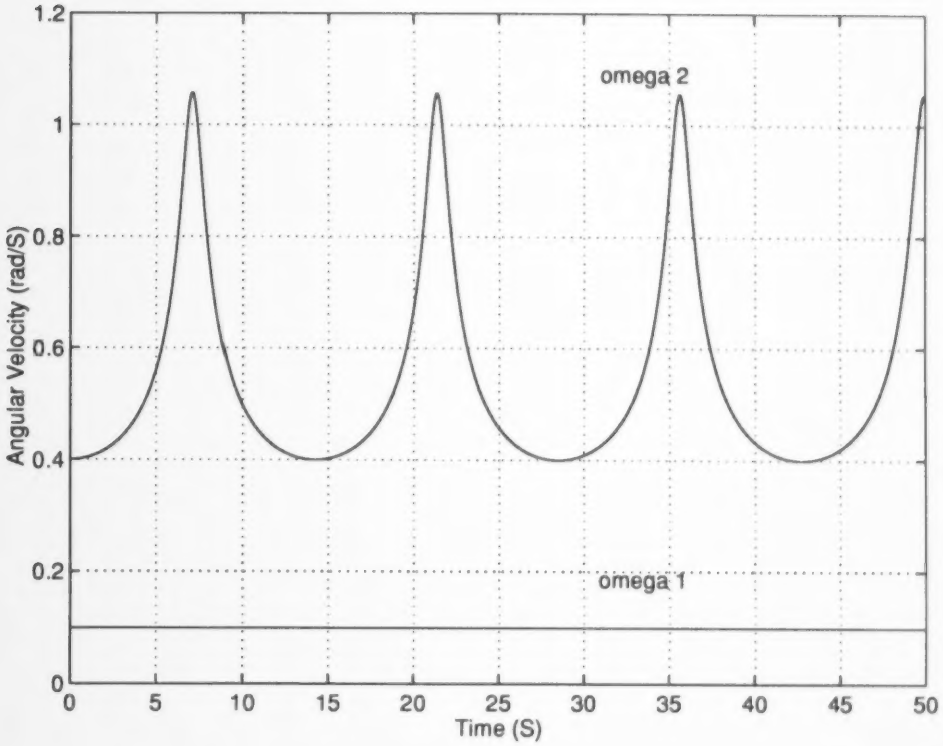


Figure 33: Graph of Link Angular Velocity

4.6 Summary

In this chapter the SIMULINK block-diagram model of a rotating link was presented. This model, created from the multibond graph model of a link, is suitable for use as a component to construct more complicated manipulator models.

The SIMULINK component block contains a derivative block, due to the presence of derivative causality in the link multibond graph model. Using the Gear stiff equation solver, simulations constructed from the SIMULINK component block are long, but nevertheless accurate.

Chapter 5

Summary and Conclusions

5.1 Summary

To provide the background necessary to understand the ideas developed in this thesis, a tutorial overview of bond graph techniques was presented in chapter 2. In addition to describing the basic multiport elements, the application of bond graph techniques to simple mechanical and electrical systems was presented, by way of example. As well, the use of the Sequential Causal Assignment Procedure (SCAP) to obtain a set of state-space equations from a bond graph model was described. The material presented in chapter 2 is contained in a number of textbooks. the most notable of these are written by Rosenberg and Karnopp (for example [22]).

In chapter 3 an extended form of bond graph, the multibond graph, was used to model the dynamics of a manipulator link. While there is a great body of literature on robot dynamics, multibond graphs, and the application

of multibond graphs to multi-body systems, the work presented in chapter 3 serves two purposes:

1. To relate multibond graph techniques to those conventionally used in developing the dynamics of manipulators.

2. To create a link model suitable for use in an object-based modeling scheme.

To accomplish these goals, a multibond graph model of a single link was created, using the kinematic structure that would be used to obtain the Newton-Euler (N-E) formulation of the link dynamics. The relationship between the multibond graph model, and the N-E approach, was demonstrated by deriving the N-E equations from an augmented multibond graph link model.

A demonstration of the utility of the multibond graph link model, in an object-based modeling environment, was presented in chapter 4. In this chapter, a SIMULINK block-diagram model of a revolute manipulator link, based on the multibond graph link model was created. Using the SIMULINK link model as a component, a planar two-link manipulator was modeled and simulated.

5.2 Conclusions

The creation of a multibond graph model of a manipulator link, using the same kinematic structure that would be used to develop the Newton-Euler formulation of the link dynamics, shows the close relationship between the

two techniques. This is somewhat surprising since bond graph techniques model dynamic systems in terms of power and energy. Thus, one would expect that the multibond graph approach was similar to one of the conventional energy-based approaches. The determining factor appears to be the point-of-view adopted in developing the dynamics, that is, whether the dynamics are developed from a local (link-based) perspective, or in terms of the entire manipulator.

The use of multibond graphs to form the basis for a object-based modeling and simulation tool for robotics has been demonstrated. The construction of a dynamic model for robotic system can be accomplished simply by connecting together multibond components corresponding to actuators, links and sensors. During construction, the detailed workings of each component is hidden. Thus, the researcher is relieved of the responsibility of being an expert in every branch of robotics.

5.3 Suggestions For Future Research

There are many applications for an object-based modeling tool in robotics. For such a tool to be effective, a library of components for actuators, sensors, and links must be available. The procedure for creating ideal models of these components is very similar to that presented here.

To design a control system, it is often necessary to make simplifying assumptions about the dynamics of a manipulator, based on an ideal model. To test the validity of these assumptions using simulation, it is necessary to have a realistic manipulator model. Thus, methods for creating more

accurate multibond graph models, perhaps using system identification techniques, should be investigated. It would be particularly interesting to see if the non-ideal effects could be isolated, and modeled separately with multiport elements. Adding or subtracting these elements from the multibond graph model would allow the designer to see exactly where the control system design broke down.

Computer animation is an excellent demonstration and teaching tool in robotics. Physically-based modeling has recently become an important research area within computer graphics. While the majority of work in this area uses object-based or object-oriented techniques, the dynamic model for each object is created using mathematical equations. It would seem that this is an ideal application for multibond graphs.

In this thesis, the simulation of multibond graphs was carried out using SIMULINK. The methods used to create the MTF block were just barely tolerated by this program; the program beeps continually while the two-link model is being loaded. The reason for this is that SIMULINK requires that all variables be initialized. In fact all the variables *were* initialized, using the masked-block function, but SIMULINK did not recognize this while loading the model. A further difficulty with the program is that the algorithm used for the Gear method is not available; it is very difficult to determine the proper step sizes for a simulation without running a series of test. Even having run those tests, the results seem strange. For example, in table 4.5 the simulation times using a step size of 10^{-2} are incredibly poor, and seem to have no relationship with the other times in the table.

Methods to improve the simulation of multibond graphs should be inves-

tigated. It would be nice to have the option of entering the multibond graph schematically, rather than having to convert it to a block-diagram form. Using the programming interface to MATLAB and SIMULINK, it should be possible to provide support for schematic entry, animation, and generation of symbolic equations. Due to the complexity of the underlying simulation engine, using the programming interface to MATLAB is preferable to developing a dedicated simulator. However, it is also possible to create a simulator based on one of the publically available MATLAB *clone* software packages. The advantage of doing this is that increased flexibility is gained by having access to the source code.

Appendix A

S-Functions for MTF Block

A.1 Connections, Appearance, and Masking for MTF Block

This M-file, generated by drawing in a SIMULINK block diagram window, contains the S-function that defines the connections, appearance and masking for the MTF block.

```
function [ret,x0,str]=mtf(t,x,u,flag);

%MTF    is the M-file description of the SIMULINK system named MTF.

%    The block-diagram can be displayed by typing: MTF.

%
```

```
%    SYS=MTF(T,X,U,FLAG) returns depending on FLAG certain
%    system values given time point, T, current state vector, X,
%    and input vector, U.
%    FLAG is used to indicate the type of output to be returned
%    in SYS.
%
%    Setting FLAG=1 causes MTF to return state derivatives, FLAG=2
%    discrete states, FLAG=3 system outputs and FLAG=4 next sample
%    time. For more information and other options see SFUNC.
%
%    Calling MTF with a FLAG of zero:
%    [SIZES]=MTF([],[],[],0),  returns a vector, SIZES, which
%    contains the sizes of the state vector and other parameters.
```

```
%         SIZES(1) number of states

%         SIZES(2) number of discrete states

%         SIZES(3) number of outputs

%         SIZES(4) number of inputs.

%    For the definition of other parameters in SIZES, see SFUNC.

%    See also, TRIM, LINMOD, LINSIM, EULER, RK23, RK45,

%    ADAMS, GEAR.


% Note: This M-file is only used for saving graphical information;

%       after the model is loaded into memory an internal model

%       representation is used.


% the system will take on the name of this mfile:
```

```
sys = mfilename;

new_system(sys)

simver(1.2)

if(0 == (nargin + nargout))

     set_param(sys,'Location',[230,130,730,430])

     open_system(sys)

end;

set_param(sys,'algorithm',        'RK-45')

set_param(sys,'Start time',    '0.0')

set_param(sys,'Stop time',        '999999')

set_param(sys,'Min step size',    '0.0001')

set_param(sys,'Max step size',    '10')

set_param(sys,'Relative error','1e-3')
```

```
set_param(sys,'Return vars',     '')



%     Subsystem  'MTF'.


new_system([sys,'/','MTF'])

set_param([sys,'/','MTF'],'Location',[220,168,672,402])


add_block('built-in/S-function',[sys,'/','MTF/S-function'])

set_param([sys,'/','MTF/S-function'],...

        'function name','mtf_int',...

        'parameters','inputString, numberOfParameters',...

        'position',[195,110,245,130])
```

```
add_block('built-in/Inport',[sys,'/','MTF/in_2'])

set_param([sys,'/','MTF/in_2'],...

        'Port','2',...

        'position',[70,115,90,135])



add_block('built-in/Outport',[sys,'/','MTF/out_1'])

set_param([sys,'/','MTF/out_1'],...

        'position',[280,110,300,130])



add_block('built-in/Inport',[sys,'/','MTF/in_1'])

set_param([sys,'/','MTF/in_1'],...

        'position',[70,100,90,120])
```

```
add_block('built-in/Mux',[sys,'/','MTF/Mux'])

set_param([sys,'/','MTF/Mux'],...

        'inputs','[3 numberOfParameters]',...

        'position',[120,100,150,135])

add_line([sys,'/','MTF'],[250,120;270,120])

add_line([sys,'/','MTF'],[95,125;110,125])

add_line([sys,'/','MTF'],[95,110;110,110])

add_line([sys,'/','MTF'],[155,120;185,120])

set_param([sys,'/','MTF'],...

        'Mask Display','MTF',...

        'Mask Type','MTF',...

        'Mask Dialogue',
```

```
        'Modulated Transformer|Modulation Matrix (function)
        |Number of Parameters',...
        'Mask Translate',' inputString = @1; numberOfParameters=@2;')
set_param([sys,'/','MTF'],...
        'Mask Help','Note: the modulation function must be
         enclosed in ''quotes'' ',...
        'Mask Entries',''''[1 0 0; 0 0 1; 0 0 1]''\/3\/')

%     Finished composite block 'MTF'.

set_param([sys,'/','MTF'],...
        'position',[230,97,260,148])
```

```
% Return any arguments.

if (nargin | nargout)

    % Must use feval here to access system in memory

    if (nargin > 3)

        if (flag == 0)

            eval(['[ret,x0,str]=',sys,'(t,x,u,flag);'])

        else

            eval(['ret =', sys,'(t,x,u,flag);'])

        end

    else

        [ret,x0,str] = feval(sys);

    end
```

```
end
```

A.2 Internal Function Called by MTF Block

```
function [sys,x0]=mtf_int(t,x,u,flag,inputString,numberOfParameters)

%

%

% mtf_int:        S-file implementation of a MTF (internal)

%                 This function is called from the masked block MTF.

%

% Description:    This function evaluates the matrix function

%                 contained in inputString, and then multiplies

%                 the resultant matrix with the input u.

%
```

```
% Note:          inputString must be enclosed in single 'quotes'.



if (abs(flag) == 0)

   % return flags

   sys = [   0                       % number of continuous states

             0                       % number of discrete states

             3;                      % number of outputs

             3+numberOfParameters    % number of inputs

             0                       % number of discontinuous roots

             1];                     % flag for direct feedthrough



elseif (abs(flag) == 3)

   % return output vector
```

```
    T = eval(inputString);              % calculate transformer matrix

    sys=[ T* [u(1);u(2);u(3)] ];        % calculate output vector

    M = size(u,1);                      % how many rows did u have?

    if( M > 1)

       sys = sys.';       % transpose if input was row vector

    end



else

   sys = [];       % don't return anything

end
```

Appendix B

Equations From Augmented Multibond Graph Link Model

The following equations are taken from the multibond graph shown in figure 25 of chapter 4.

$$e_5 = \dot{H}_i \tag{B.1}$$

$$f_5 = ({}^0J_i)^{-1}H_i \tag{B.2}$$

$$e_5 = e_4 - e_6 - {}^0N_{i+1} - e_7 - e_12 \tag{B.3}$$

$$f_5 = f_4 = f_6 = f_7 = f_{12} = f_{18} \tag{B.4}$$

$$e_6 = \Lambda_i {}^0J_i f_6 \tag{B.5}$$

$$e_2 = e_3 = e_4 \tag{B.6}$$

$$f_2 = f_4 - {}^0\omega_{i-1} \tag{B.7}$$

$$e_1 = {}^0\tau_i = ({}^0R_{i-1})^T e_2 \tag{B.8}$$

$$f_1 = ({}^0R_{i-1})^T f_2 \tag{B.9}$$

$$e_7 = (R_i^*)^T e_8 \tag{B.10}$$

$$f_8 = R_i^* f_7 \tag{B.11}$$

$$e_{12} = (\bar{S}_i)^T e_{13} \tag{B.12}$$

$$f_{13} = \bar{S}_i f_{12} \tag{B.13}$$

$$e_8 = e_9 = e_{10} \tag{B.14}$$

$$f_{10} = {}^0v_{i-1} + f_8 \tag{B.15}$$

$$e_{10} = e_{11} + {}^0F_{i+1} \tag{B.16}$$

$$f_{10} = f_{11} = f_{17} \tag{B.17}$$

$$e_{11} = e_{13} = e_{14} \tag{B.18}$$

$$f_{14} = f_{11} + f_{13} \tag{B.19}$$

$$e_{14} = e_{15} + e_{16} \tag{B.20}$$

$$f_{14} = f_{15} = f_{16} \tag{B.21}$$

$$e_{15} = M_i \dot{f}_{15} \tag{B.22}$$

$$e_{16} = \Lambda_i M_i f_{16} \tag{B.23}$$

Appendix C

S-Function for Two-link Manipulator

Note: Due to the size of the listing file, this appendix is contained im a supplemental volume. The listing code was generated by SIMULINK, and is only of interest for verifying the physical parameters of the example two-link manipulator in chapter 4.

Bibliography

[1] R. R. Allen. Dynamics of mechanisms and machine systems in accelerating reference frames. *Transactions of the ASME: Journal of Dynamic Systems, Measurement, and Control*, 103:395–403, December 1981.

[2] Farid M. L. Amirouche. *Computational Methods in Multibody Dynamics.* Prentice-Hall Inc, A Simon & Shuster Company, Englewood Cliffs, New Jersey 07632, 1992.

[3] Myriam Beauwin and Francis Lorenz. Representation of 3-d mechanisms using 6-dimensional multibonds. In R. Vichnevetsky, P. Borne, and J. Vignes, editors, *12th IMACS World Congress, July 18-22 1988,Paris,France*, 1988.

[4] Wayne J. Book. Modeling, design, and control of flexible manipulator arms: A tutorial review. In *Proceedings of the 29th Conference on Decision and Control, Honolulu, Hawaii*, pages 500 – 505. IEEE, 1990.

[5] A. M. Bos. *Modeling Multibody Systems in Terms of Multibond Graphs.* PhD thesis, University of Twente, Enschede, The Netherlands, 1986.

[6] P. C. Breedveld. Proposition for an unambiguous vector bond graph notation. *Transactions of the ASME: Journal of Dynamic Systems, Measurement, and Control*, 104:267–270, September 1982.

[7] Peter C. Breedveld. Decomposition of multiport elements in a revised multibond graph notation. *Journal of the Franklin Institute*, 318(4):253–273, October 1984.

[8] Peter C. Breedveld. Decomposition of multiport elements in a revised multibond graph notation. *Journal of the Franklin Institute*, 319(1/2):1–36, January/February 1985.

[9] John J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison-Wesley Publishing Company, Inc., 1989.

[10] J. Van Dijk and P. C. Breedveld. Simulation of system models containing zero-order causal paths—i. classification of zero-order causal paths. *Journal of the Franklin Institute*, 328(5/6):959–979, 1991.

[11] J. Van Dijk and P. C. Breedveld. Simulation of system models containing zero-order causal paths—ii. numerical implications of class 1 zero-order causal paths. *Journal of the Franklin Institute*, 328(5/6):981–1004, 1991.

[12] Jenny Montbrun-Di Filippo, Marisol Delgado, and and others. A survey of bond graphs: Theory, applications and programs. *Journal of the Franklin Institute*, 328(5/6):565–606, 1991.

[13] Grant R. Fowles. *Analytical Mechanics.* CBS College Publishing, 1986.

[14] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee. *Robotics:Control Sensing, Vision and Intelligence.* McGraw-Hill Inc., 1987.

[15] Herbert Goldstein. *Classical Mechanics: Second Edition.* Addison-Wesley, 1980.

[16] William H. Hayt Jr. and Jack E. Kemmerly. *Engineering Circuit Analysis.* McGraw-Hill, 1978.

[17] Chris Malcolm, Tim Smithers, and John Hallam. An emerging paradigm in robotic architecture. In *Intelligent Autonomous Systems 2: Proceedings of an International Conference held in Amsterdam, the Netherlands*, pages 546–563. Stichting International Congress of Intelligent Autonomous Systems, 1990.

[18] The MathWorks Inc. *SIMULINK 1.2 User's Guide*, 1992.

[19] M. Andrew Moshier and Edward F. Sowell. Specifying dynamic models in the simulation problem analysis kernel. In Vijay Madisetti, editor, *Modeling and Simulation on Microcomputers, 1990: Proceedings of the SCS Multiconference on Modeling and Simulation on Microcomputers, 17-19 January, 1990, San Diego, California*, pages 117–124. Society for Computer Simulation (Simulation Council, Inc.), San Diego, California, 1990.

[20] Eugene I. Rivin. *Mechanical Design of Robots.* McGraw-Hill, 1988.

[21] R. C. Rosenberg and D. C. Karnopp. A definition of the bond graph language. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 94(3):179–182, 1972.

[22] R. C. Rosenberg and D. C. Karnopp. *Introduction to Physical System Dynamics*. McGraw-Hill, 1983.

[23] M. Shahinpoor. *A Robot Engineering Textbook*. Harper & Row, New York, 1987.